DATREND Systems Inc.

vPad-Check™

Automated Maintenance & Inspection App for vPad Safety Analyzers

Operating Manual

vPad-Check™

Automated Maintenance and Inspection App for vPad Safety Analyzers

Operating Manual

© 2013 Datrend Systems Inc.

Unit 130 - 4020 Viking Way
Richmond, BC • CANADA • V6V 2L4
Tel: 800.667.6557 (North America Only) or
604.291.7747 • Fax 604.294.2355
e-mail: **customerservice@datrend.com**

Revision History		
Revision	Description	Date
A	Initial release	2013-Dec-12

Copyright

Datrend Systems Inc. (“DSI”) agrees to a limited copyright release that allows you to reproduce manuals and other printed materials for use in service training programs and other technical publications. If you would like other reproductions or distributions, submit a written request to Datrend Systems Inc.

Unpacking and Inspection

Follow standard receiving practices upon receipt of goods. Check the shipping carton for damage. If damage is found, stop unpacking. Notify the freight carrier and ask for an agent to be present while the goods are unpacked. There are no special unpacking instructions, but be careful not to damage the goods when unpacking. Inspect the goods for physical damage such as bent or broken parts, dents, or scratches.

Claims

Our routine method of shipment is via common carrier. Upon delivery, if physical damage is found, retain all packing materials in their original condition and contact the carrier immediately to file a claim. If the goods are delivered in good physical condition but does not operate within specifications, or if there are any other problems not caused by shipping damage, please contact your local sales representative or DSI immediately.

Standard Terms and Conditions

Refunds & Credits

Please note only serialized products (products labelled with a distinct serial number) and accessories are eligible for partial refund and/or credit. Non-serialized parts and accessory items (cables, carrying cases, auxiliary modules, etc.) are not eligible for return or refund. In order to receive a partial refund/credit, the product must not have been damaged, and must be returned complete (meaning all manuals, cables, accessories, etc.) within 90 days of original purchase and in “as new” and resalable condition. The *Return Procedure* must be followed.

Software which has been activated may not be returned for credit. Use it in Demo mode where possible to ensure suitability for purpose.

Return Procedure

Every product returned for refund/credit must be accompanied by a Return Material Authorization (RMA) number, obtained from Datrend Customer Service. All items being returned must be sent *prepaid* (freight, duty, brokerage, and taxes) to our factory location.

Restocking Charges

Product returned within 30 days of original purchase are subject to a minimum restocking fee of 15%. Product returned in excess of 30 days after purchase, but prior to 90 days, are subject to a minimum restocking fee of 20%. Additional charges for damage and/or missing parts and accessories will be applied to all returns. Product which is not in “as new” and resalable condition, are not eligible for credit return and will be returned to the customer at their expense.

Certification

This instrument was thoroughly tested and inspected and found to meet DSI’s specifications when it was shipped from the factory.

Where applicable, calibration measurements are traceable to the National Research Council of Canada (NRC) and/or the National Institute of Standards and Technology (NIST). Devices for which there are no NRC/NIST calibration standards are measured against in-house performance standards using accepted test procedures.

Warranty and Product Support

Software Applications (Apps) for use in conjunction with ES601 and vPad Safety Analyzers and other Datrend Products are supplied subject to the Terms and Conditions agreed to by the act of Activating the App. In the event no Activation is required, the Terms and Conditions published on the Datrend Systems Inc. (DSI) website shall prevail. By using the App, you agree to these terms and Conditions. IN NO EVENT SHALL DATREND SYSTEMS INC. BE LIABLE FOR CONSEQUENTIAL DAMAGES.

DSI DISCLAIMS ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR APPLICATION.

vPad™, vPad-XPORT™, vPad-ES™, vPad-353™, vPad-AS™, vPad-A3™,vPad-Record Manager™, vPad-RM™ , vPad-EQM™, vPad-Check™, vPad-Cal™, and Datrend Docs™ and CMX**™** are trademarks of Datrend Systems Inc.

Android™ is trademark of Google Inc.

Table of Contents

TABLE OF CONTENTS III

1 OVERVIEW 1
1.1 Checklists 1
1.1.1 Description 1
1.1.2 Operation 2
1.2 vPad Programming Language 3

2 INSTALLATION 5

3 OPERATION 7
3.1 Startup 7
3.2 Main Menu 9
3.2.1 Checklist Search 10
3.2.2 Update Checklist Files 10
3.2.3 Label Printer Setup 11
3.2.3.1 Printer Connection 11
3.2.3.2 Printer Setup Menu 14
3.3 Running a Checklist 16
3.3.1 Equipment Information 17
3.3.2 Prompts 19
3.3.3 Images and Video 20
3.3.4 Checks 21
3.3.4.1 Pass-Fail Inspection 21
3.3.4.2 Pass-Fail With User Data Input 22
3.3.5 Limit Tests 23
3.3.6 ECG Tests 24
3.3.7 Electrical Safety Tests 25
3.3.8 Remote Control 30
3.3.8.1 Remote Device Connection 30
3.3.8.2 Remote Control Steps 34
3.3.9 Checklist Results Preview 36
3.3.10 Creating a Test Record 37
3.3.11 Printing an Inspection Label 41
3.4 Using the Android Keyboard 42

4 vPAD PROGRAMMING LANGUAGE 45
4.1 Checklist Files 45
4.1.1 File Location 45
4.1.2 Filenames 45
4.1.3 File Editing 46
4.2 Checklist Structure 48

4.2.1 Blank Lines 48
4.2.2 Comments 48
4.2.3 Statements 48
4.2.4 Keywords 49
4.2.5 Strings 49
4.2.6 Newline Character Sequence 50
4.2.7 Line Continuation Sequence 50
4.3 prompt Statement 52
4.4 show Statement 54
4.5 check Statement 55
4.6 limtst Statement 56
4.7 color Statement 59
4.8 power Statement 60
4.9 ecgsim Statement 61
4.10 run Statement 64
4.11 Remote Control Statement 65
4.11.1 xopen Statement 65
4.11.2 xctrl Statement 67
4.11.3 xtest Statement 69

5 UPGRADES AND MAINTENANCE 71

1 Overview

vPad-Check™ is a software application (App) which may be installed on vPad-ES™ and vPad-353™ electrical safety analyzers, and which may be used to automate preventative maintenance (PM) inspections, performance testing and other types of service performed on medical equipment. vPad-Check automation provides these benefits to PM's and similar technical work:

- Reduced job time
- Increased labour efficiency
- Increased consistency of work product - procedures are performed the same way each time, with no missing steps
- Automatic recording of work performed and results obtained - Test Record files provide documented evidence that equipment was tested and was found acceptable

This chapter provides an overview of vPad-Check and provides an introduction to PM inspection Checklists and the tool used to create them, the vPad Programming Language (vPL).

1.1 Checklists

1.1.1 Description

vPad-Check™ implements procedure automation through ***Checklists***. A Checklist may be thought of as an inspection or maintenance procedure broken down into a list of steps which are to be performed in a certain order on the equipment under service (EUS). A step of a Checklist may:

- Display instructions to the technician, ranging from a single line of text to detailed paragraphs
- Display a drawing or photograph of the EUS
- Play a video showing setup or operation of the EUS

- Prompt the technician to inspect an item or feature of the EUS, record details, and assign a "PASS" or "FAIL" to the inspected item

- Prompt the technician to perform a test or a measurement and enter the result, which is then passed or failed automatically according to quantitative criteria included in the Checklist

- Execute an electrical safety test of the EUS automatically

- Execute a sequence of test waveforms automatically (useful for checking patient monitors, defibrillators and other medical equipment incorporating an ECG function)

- Execute performance tests of the EUS automatically, by means of external analyzers and simulators interfaced through serial or wireless connections to the vPad

1.1.2 Operation

To perform a PM or other service procedure on a medical device, the technician begins by selecting the required Checklist from a scrollable list. In the list, Checklists are typically named according to the type of device being serviced (e.g. anesthesia unit, defibrillator, infusion pump and so on), although it is also possible for there to be Checklists which are specific to particular models (e.g. HP CodeMaster defibrillator).

After selecting the appropriate Checklist, the technician will step through the procedure one step at a time, pausing to read instructions, review drawings or photographs, inspect items, and enter data as required by the procedure. Many Checklists will incorporate an electrical safety test of the equipment under service (EUS), which will be executed automatically by either the vPad-AS™ or vPad-A3™ software application installed on the tablet. Checklists may also include performance tests, which may be done manually by the technician and then entered into vPad-Check, or automatically by externally-connected test instrumentation. Some Checklists, for example a PM of an anaesthesia machine, may be complex and there is no practical limit as to the number of steps that a Checklist may have.

If necessary, the technician can step backward through the Checklist to repeat certain tests, or to review or re-enter data that was input earlier in the procedure. Steps requiring the technician to perform a pass/fail assessment of an inspection item, or a test or measurement, can be "skipped" or assigned a status of "not applicable". Any procedure step which has been "skipped" or assigned "not applicable" status will be saved and reported as such in the Test Record.

After completing the procedure, a record of the service including all pass/fails, measurements and results will be saved in the vPad filing system. In addition to these results, a Test Record may include equipment information entered by the technician during the procedure, such as make, model, serial number, location and so on. In order to create a Test Record, at minimum an equipment control number or other ID must be entered. Test Record files are named according to this entered ID or control number.

Once the Test Record has been saved, a PM inspection label may then be printed by means of a (optional) label printer connected to the vPad. vPad-Check supports PM label printing with several different printer models. To make use of this feature, the user must have a compatible printer and the corresponding Smart Cable adaptor for interfacing the printer to vPad. More information on PM label printing may be found in Sections 3.2.3 and 3.3.11 of this manual.

1.2 vPad Programming Language

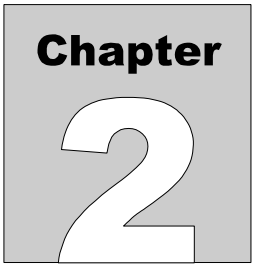
vPad "Checklists" correspond to a set of files which are stored in the vPad-Check directory of the Android tablet. Each Checklist file has a "**`.vpl`**" file extension and contains a ***program*** which has been written in the vPad Programming Language or "***vPL***".

Each step of a Checklist procedure has a direct one-to-one relationship with a corresponding vPL ***statement*** in the Checklist file. vPL provides statements for displaying instructions, showing photographs and video, accepting data input from the user, running electrical safety tests, interfacing with external test instruments and so on.

vPL is a simple scripting language which does not include branch or loop constructs. As such, vPL is a straightforward language that lends itself to easy creation and maintenance of PM's and other service procedures with minimal learning curve.

Checklist files may be edited with any text editor or word processor. Minor changes or additions to a Checklist may be made using the text editor app installed on the tablet. Wholesale modification or creation of new Checklists is more efficiently accomplished with a personal computer (PC). When using a PC for Checklist editing or creation, file(s) may be transferred from the PC to the tablet by USB flash drive, WiFi network, or point-to-point Bluetooth. vPad-Check provides an "Update Files" feature for this purpose.

More information on the vPad Programming Language may be found in Chapter 4 of this manual.



2 Installation

The vPad-Check App is normally pre-installed on the vPad tablet, but the icon for the App may not have been placed on the Home screen .

Go to the Apps screen by pressing the Apps icon, , in the upper right corner of the Home screen. Find the vPad-Check icon and press and hold the icon for a few seconds. You will then be able drag and drop the icon onto the Home screen.

In order to use the App it will need to be activated. When the App is started for the first time, the activation screen will be presented. Enter the activation code provided by your dealer to allow the App to run.

In the event that vPad-Check is not on the tablet, contact your dealer and an installation package for the app will be provided via download.

In order to use vPad-Check, you must have at least one software application installed and activated for automated electrical safety testing. For testing per AAMI-ES1 and/or IEC60601 standard, the vPad-AS™ App must be installed and activated. For testing per IEC62353, the vPad-A3™ App must be installed and activated.

Chapter 3

3 Operation

This chapter covers operation of the vPad-Check app and Checklist execution.

3.1 Startup

Press the vPad-Check icon,

, to start the App. A splash screen will display the product name and the software version.

Figure 1 *- vPad-Check startup screen.*

As the splash screen is displayed, the app will "connect" to the Analyzer Base Unit of the vPad and cause it to perform a number of internal self-tests. The AC supply which powers the Analyzer Base Unit is also tested. Normally, all tests will complete successfully and the Checklist selection screen will then be displayed.

Faults in the Base Unit or the AC supply may be detected during the self-test, in which case an alarm message is displayed below the software version number. In the event of a Base Unit fault, the app will automatically exit to the Home screen. In the event of an AC supply fault, in addition to the alarm message the app will display an Alert Dialog, which you must then acknowledge by pressing "Continue", "Exit"or "OK" (see ***Figure 2***). Refer to TABLE 3.1-1 for recommended actions in the event of a startup alarm message.

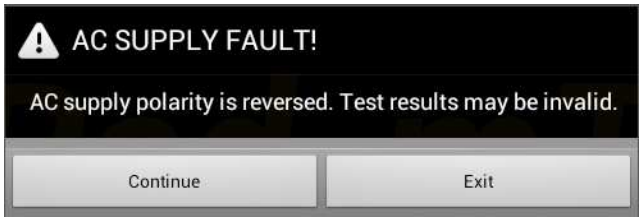


Figure 2 *- Alert Dialog: AC supply fault.*

TABLE 3.1-1: Startup Alarms and Corrective Actions

Alarm Message	Cause	Actions
ERROR: cannot communicate with Base Unit	App is unable to send messages to one or more instrument systems inside the Base Unit	• Check Base Unit has power • Check Base-to-tablet USB connection • If AC power and USB cables are OK, press and hold RESET button on left side of Base Unit, then try restarting the app • If error persists, contact Datrend Customer Service (see Chapter 5 for contact information)
AC SUPPLY FAULT (polarity reversed)	AC outlet incorrectly wired	• Do not use vPad with incorrect or unsafe AC supply • Connect Base Unit to a different outlet
AC SUPPLY FAULT (ground disconnected)	AC outlet incorrectly wired	• Do not use vPad with incorrect or unsafe AC supply • Connect Base Unit to a different outlet
AC SUPPLY FAULT (ground disconnected)	Isolated power system	• Do not perform leakage testing as results will be invalid • Optionally, vPad may be used to perform tests other than leakage (e.g., voltage, resistance or ECG simulator tests) • If possible, change to grounded power outlet
RMS zero calibration failure	Unable to zero voltmeter	• Press RESET button on left side of Base Unit for 3 seconds, then try restarting app • If problem persists, contact Datrend Customer Service (see Chapter 5 for contact information)

3.2 Main Menu

Following a successful start-up, vPad-Check will display the main menu of the App:

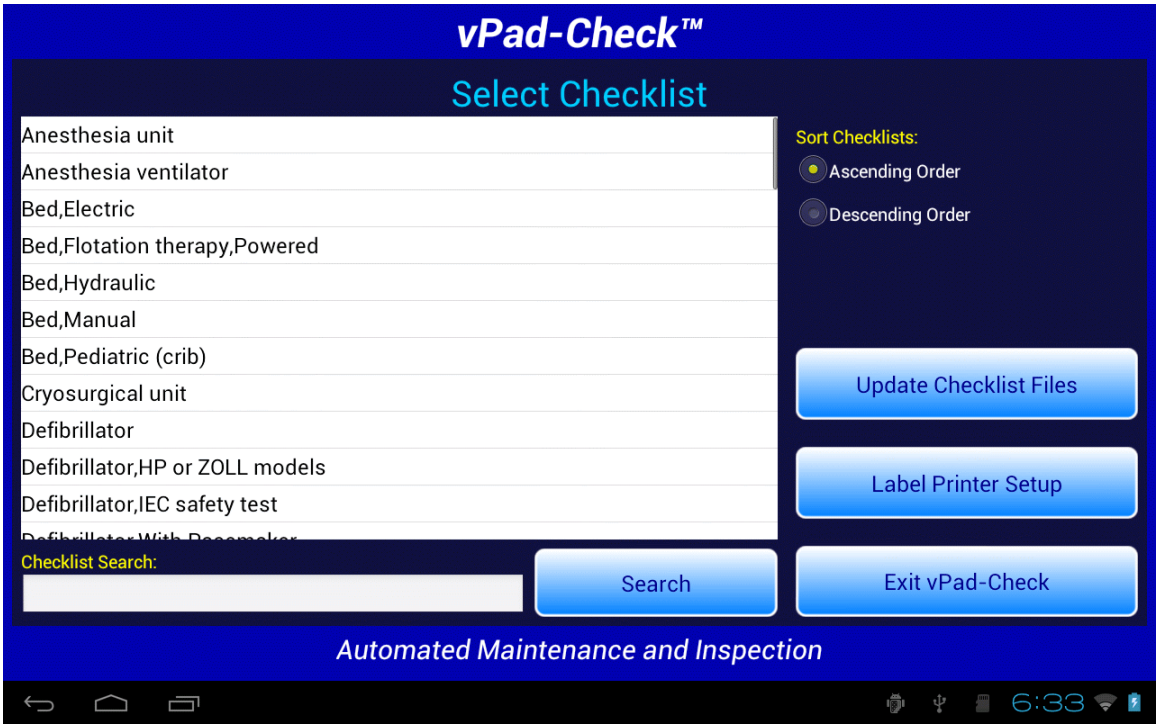


Figure 3 *- Main menu of vPad-Check.*

Some example Checklists are supplied with the App and will appear in the scrollable list as shown above. Example Checklists include PM procedures for typical medical devices as well as some selected models of specific devices. The listing may be sorted in ascending or descending order by selecting the corresponding radio button at upper right on the screen.

To start a test, press and drag the list view until the desired Checklist is found, then tap on the Checklist. Alternatively, you may start a test by entering the complete name of a Checklist in the "Checklist Search" box and pressing "**Search**".

To return to the Android Home screen, press the icon on the navigation bar. Note this will only pause the App temporarily and will not actually terminate or close vPad-Check. To fully close the App, press the "**Exit vPad-Check**" button or alternatively the Android "BACK" navigation button when viewing the main menu.

3.2.1 Checklist Search

To search for a particular Checklist, tap the "**Checklist Search**" box of the main menu (***Figure 3***) , enter the first few characters of the Checklist name, then press the "**Search**" button. This will move the list to the nearest matching item, which will then appear highlighted in green. If the full name of the Checklist is entered, and an exact match to the entered name is found, then the Checklist will begin execution automatically.

3.2.2 Update Checklist Files

vPad-Check provides a file-update feature which is intended to facilitate the transfer of Checklist files (files with a "**`.vpl`**" extension) from a personal computer (PC) to the Android tablet. This feature is helpful if you will be using a PC word processor or text editor to write your own Checklists.

Checklist files may be easily transferred by USB flash drive. Copy files on the PC to the root directory of a USB flash drive, then connect the flash drive to the USB "HOST" port of the tablet. Depending on the tablet model, an adaptor cable provided with the tablet may be required for the flash drive connection (***Figure 4***).

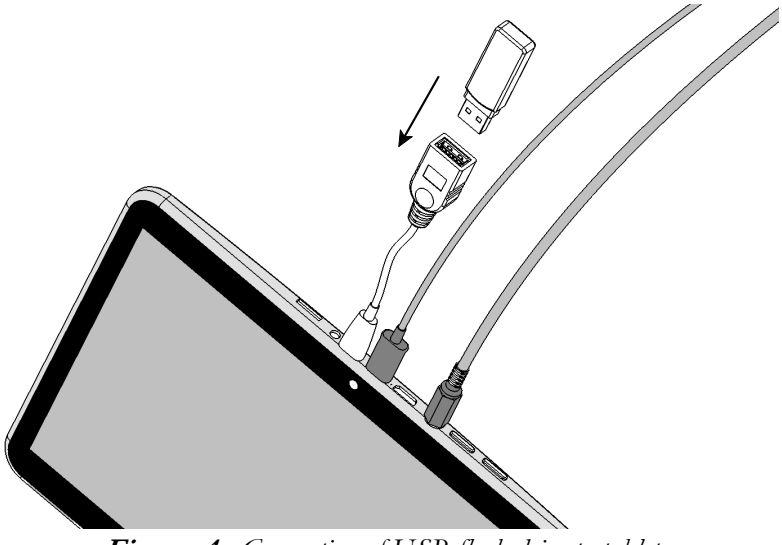


Figure 4 *- Connection of USB flash drive to tablet.*

After connecting the USB drive, press the "**Update Checklist Files**" button on the main menu of the App (***Figure 3***). vPad-Check will detect the attached USB drive and will then search for "**`.vpl`**" files in the root directory of the drive. If any such files are found, they are copied to the vPad-Check directory of the tablet. Files on the tablet which have the same filename as those on the USB drive will be overwritten.

Alternatively, Checklist files may be updated through Bluetooth transfer. If there are no files found on a USB drive when "**Update Checklist Files**" is pressed, then vPad-Check will search for "**`.vpl`**" files in **`Bluetooth`** file transfer directories of the tablet (there are several directories that could potentially be used for such transfers, depending on the software employed in transmitting files from PC to tablet).[1]

An error dialog will be displayed in the event that vPad-Check is unable to find any "**`.vpl`**" files in the expected USB or Bluetooth directories:

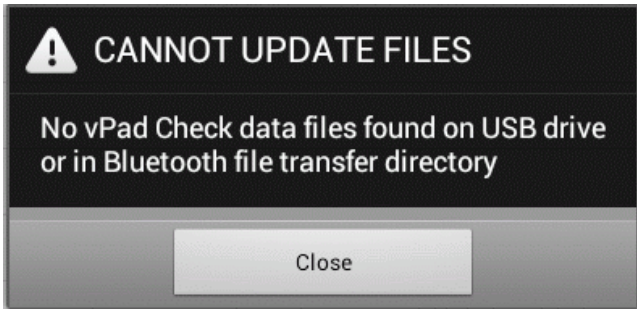


Figure 5 - *File update error message.*

3.2.3 Label Printer Setup

Once a Test Record has been saved at the end of a test, you have the option of printing an inspection label for application to the equipment under service. This "PM label" will include the ID or control number of the equipment, pass/fail status, the test date, the date at which the next inspection or PM becomes due, and a technician ID.

This section explains how to set up a label printer and configure vPad-Check for label printing.

3.2.3.1 Printer Connection

vPad-Check is compatible with label printer models listed in TABLE 3.2-1. To interface a label printer to the vPad, an XBUS Smart Cable or "***XSC***" interface will be required; refer to the table for corresponding Datrend part numbers. XSC adaptors are available for wired or wireless interface to the printer, with the wireless option being capable of both wired and wireless operation.

---

1 Bluetooth file transfers are managed by specific software which may be run on your personal computer, and which allows a user to select one or more files and "push" them via Bluetooth to a wireless mobile device such as a phone or tablet. Operation of third-party PC software used for Bluetooth file transfer is outside the scope of this operator manual; refer to instructions provided with your PC for Bluetooth-related features and installed software applications.

TABLE 3.2-1: Compatible Label Printers

Printer Make/Model	XSC Interface, Wired (serial only)	XSC Interface, Wireless (Bluetooth & serial)
Dymo LabelWriter SE300 or SE450	7400-100	7400-098
Brady TLS PC Link	7400-102	7400-101

Figure 6 shows the general setup for a wired connection to a label printer. The figure is a schematic representation and is not drawn to scale. Setup instructions are as follows:

a. Using the RJ11-6 cable provided with the XSC interface P/N 7400-100 (Dymo) or 7400-102 (Brady), connect the "XBUS" Port of the vPad Base Unit to either of the "**XBUS**" inputs of the interface. Note that one of the XBUS inputs will remain available to permit daisy-chaining of additional XSC units for remote device interfacing.

b. Connect the serial port of the label printer to the "**DEVICE**" output of the XSC interface.

c. Connect the power supply provided with the printer to the printer's power input.

d. Apply power to the printer when required. The XSC adaptor receives its power from the vPad Base Unit, through the XBUS connection.

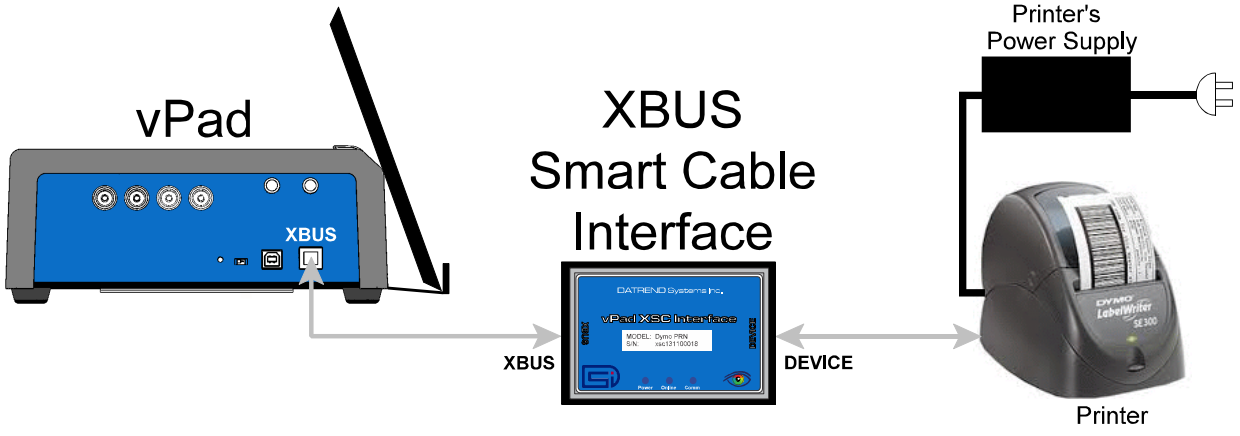


Figure 6 *- Schematic showing setup of label printer - wired (serial) interface.*

Figure 7 shows the general setup for a wireless connection to a label printer. The figure is a schematic representation and is not drawn to scale. Communication is based on point-to-point Bluetooth and is possible over distances of up to thirty feet (10 meters).

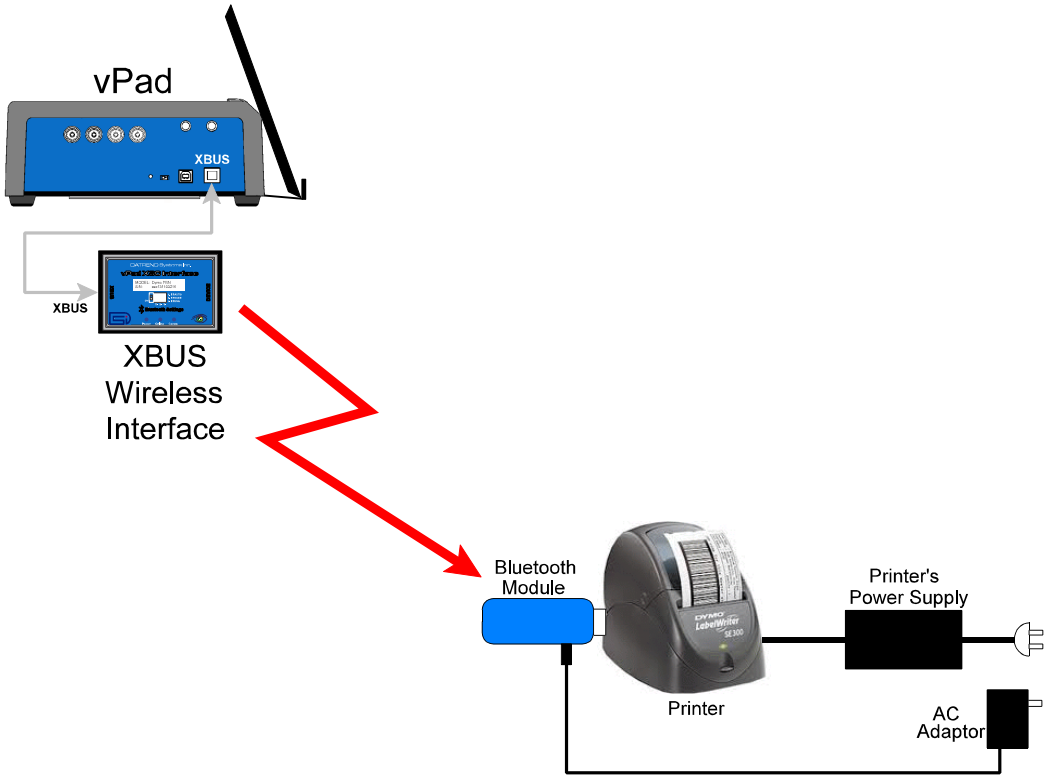


Figure 7 - *Schematic showing setup of label printer - wireless (Bluetooth) interface.*

Wireless setup instructions are as follows:

a. Using the RJ11-6 cable provided with the XSC interface P/N 7400-098 (Dymo) or 7400-101 (Brady), connect the "XBUS" Port of the vPad Base Unit to either of the "**XBUS**" inputs of the interface. Note that one of the XBUS inputs will remain available to permit daisy-chaining of additional XSC units for external device interfacing.

b. Connect the Bluetooth Module provided with the XSC interface to the serial port of the printer. Note this Bluetooth Module has been configured to communicate specifically with its companion XSC interface.

c. Connect the AC adaptor provided with the XSC interface to the DC power input of the Bluetooth Module, then connect the adaptor to the AC supply.

d. Connect the power supply provided with the printer to the printer's power input.

e. Apply power to the printer when required. The XSC interface receives its power from the vPad Base Unit, through the XBUS connection. When powered, the XSC interface will seek out the Bluetooth Module at the printer, and if the Module is powered up, a point-to-point radio link will be established automatically.

f. Optionally, a wireless XSC may be switched to wired operation. To do so, disconnect the XSC from the vPad and set the "BTENA" switch of the interface to the "OFF" position. This will disable the Bluetooth connectivity of the XSC. A wired connection to the printer may then be set up as shown in ***Figure 6***.

3.2.3.2 Printer Setup Menu

To configure vPad-Check to work with a connected label printer, press the "**Label Printer Setup**" button on the main menu of the App (see ***Figure 3***). The printer setup menu will then be shown:

Figure 8 *- Printer Setup menu.*

The printer setup menu provides the following functions:

Enable label printing	If checked, the technician will be prompted to print a PM inspection label after a Test Record has been saved. Default is *not* checked.
Printer type	Select from these models: Dymo SE300, Dymo SE450 or Brady TLS PC Link
Label title	Enter a "title"(e.g., name of institution or service agency), which will be printed in bold font at the top of the label. This feature is enabled only for Dymo model printers, which print on larger label stock than Brady.
Next test in	Sets a default interval for a "next test due" date which is printed on the label. Initially, the due date is set to the date of the test (i.e., 'today') plus the "next test" interval from the printer setup menu. This initial due date may then be adjusted if necessary, when the Test Record is saved. A "next test due" date may also be provided to vPad-Check by other automation Apps such as vPad-EQM™, which will override this default schedule.
Test Printer	Causes the printer to print a test label.
Save & Exit	Save settings and exit the menu.
Cancel	Exit the menu without changing the settings.

3.3 Running a Checklist

A test is started by selecting the required Checklist from the main menu of vPad-Check (see ***Figure 3***).

A Checklist begins by prompting you to enter information about the equipment under service. After this, you will step through the procedure one step at a time, pausing to read instructions, inspect items, and enter data as required by the procedure. Electrical safety tests and device-specific performance tests may also be executed during the procedure. At the end of the Checklist, a Test Record is saved and an inspection label may be optionally printed.

Checklists are made up of individual steps, where a step may be any of the following:

Step Type	Purpose or Function
Prompt	Displays information or instructions.
Show	Displays a drawing, photo or video.
Check	Prompts you to inspect an item and enter an assessment of the item (for example, a 'pass' or a 'fail'). Optionally, test data, comments or information about the item may also be input.
Limit Test	Prompts you to perform a test or a measurement of the equipment under service and enter the result, which is then passed or failed by vPad-Check per requirements included in the Checklist.
Colour	Sets the foreground colour of text displayed by prompts, checks and limit tests.
Electrical Safety Test	Executes an automated electrical safety test, by launching either the vPad-AS™ app or the vPad-A3™ app to perform the test.
ECG Test	Controls the vPad ECG simulator; technician is prompted to enter results based on observation of the equipment under test.
Receptacle Control	Applies power to the equipment under service by means of the vPad test receptacle.
Remote Control	Depending on the step, performs one of the following operations: • opens a connection to a remote device through a XBUS Smart Cable (XSC) interface • controls a remote device through an XSC • captures a measurement value or a test result from a remote device through a XSC

Staring with Section 3.3.2, this chapter will provide some illustrative examples of these Step Types for the case of the "Demo Checklist" included with vPad-Check, and will explain operation of the user interface in each case.

3.3.1 Equipment Information

When a Checklist is started, the first screen shown will be for entering information about the equipment:

Figure 9 *- Equipment Information menu.*

Tap on a text field to display the Android keyboard and enter information. If an external keyboard is connected to the tablet, the Android keyboard is not displayed and data entry is then accepted from the external input device. A barcode scanner can also be used for data entry. Refer to Section 3.4 for detailed instructions on using the Android keyboard.

Figure 10

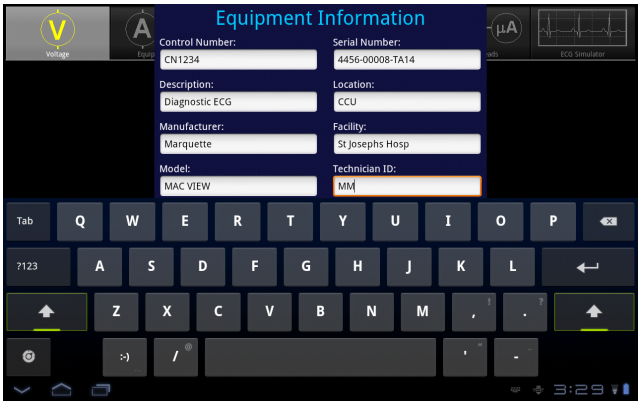


Figure 11

Pressing ENTER on the keyboard advances data input to the next text field on the screen.

Alternatively, use to navigate the text fields, or simply tap a field to jump immediately there.

Note: At a minimum, an equipment control number, or other ID, must be entered in the **Control Number** field in order to save a Test Record.

"**Facility**" and "**Technician ID**" fields are backed up in permanent storage on the Android tablet. These information fields may be modified, but the last values are maintained after power to the vPad is switched off, and are restored on subsequent use of the App.

When done entering information, press to dismiss the keyboard and expose the following buttons at the bottom of the menu:

Escape	Cancel the Checklist and return to the main menu. No test results are saved.
Next Step	Press to proceed with the next step of the Checklist.

'Escape' and **'Next Step'** are provided on many screens of the Checklist, where they provide the functions described above. Most screens also provide a **'Previous Step'** button, which causes the Checklist to go backward one step.

3.3.2 Prompts

A typical Checklist will include a number of prompt steps. Prompts are used for displaying information or instructions. Different fonts may be used for prompts, depending on the amount of information to be shown.

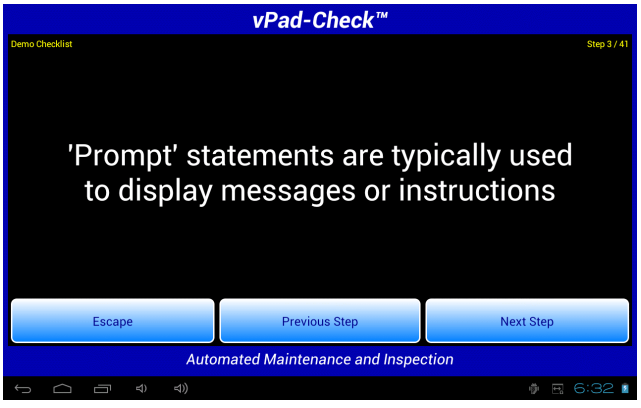


Figure 12 *- Prompt with 'normal' font.*

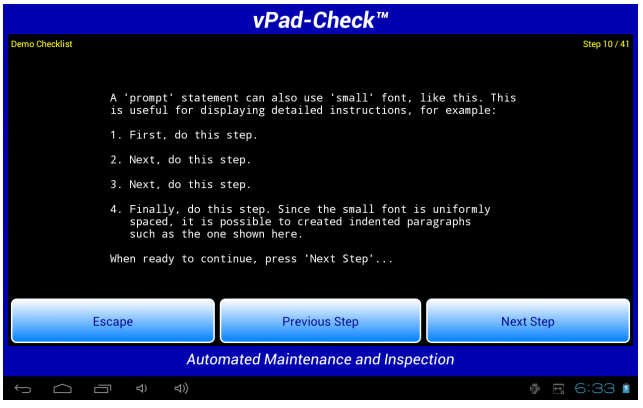

vPad-Check™
Demo Checklist
Step 10 / 41
A 'prompt' statement can also use 'small' font, like this. This is useful for displaying detailed instructions, for example:
1. First, do this step.
2. Next, do this step.
3. Next, do this step.
4. Finally, do this step. Since the small font is uniformly spaced, it is possible to created indented paragraphs such as the one shown here.
When ready to continue, press 'Next Step'...
Escape
Previous Step
Next Step
Automated Maintenance and Inspection
6:33

Figure 13 *- Prompt with 'small' font.*

Prompt steps provide **'Escape'**, **'Next Step'** and **'Previous Step'** buttons which function as described in Section 3.3.1. A prompt may also have a timeout. In this case, a timer will appear at the top of the screen.

Figure 14 *- Prompt with timer.*

The Checklist will advance to the next step automatically when the countdown reaches zero. Alternatively, the **'Next Step'** button can be pressed to advance the Checklist before the timeout expires.

3.3.3 Images and Video

A Checklist may include steps for showing drawings or photographs, or for playing previously-recorded video. Such steps will cause vPad-Check to display a description with a **'Show Picture'** or **'Play Video'** button below the text.

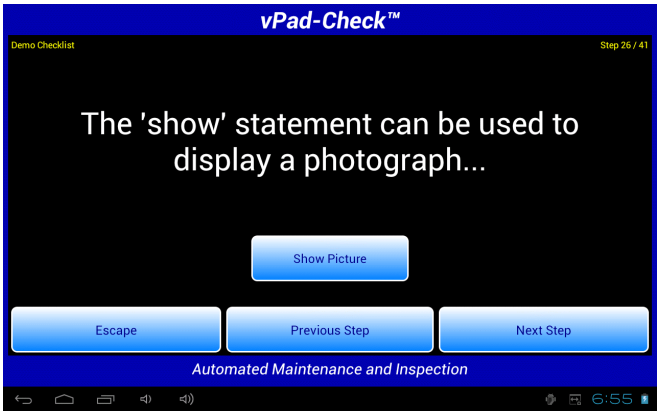


Figure 15 *- Checklist step for showing a photo or drawing.*

An Android device will usually provide a number of different Apps that are capable of displaying photos or video. When the **'Show Picture'** or **'Play Video'** button is pressed, you may see a "**Complete action using...**" menu which asks you to choose the installed App you would prefer to use. Choose an App, and if you are familiar with this App, also check the "**Use by default...**" box so that this pop-up menu will not be shown again. The image or video from the Checklist step will then be displayed by the selected App.

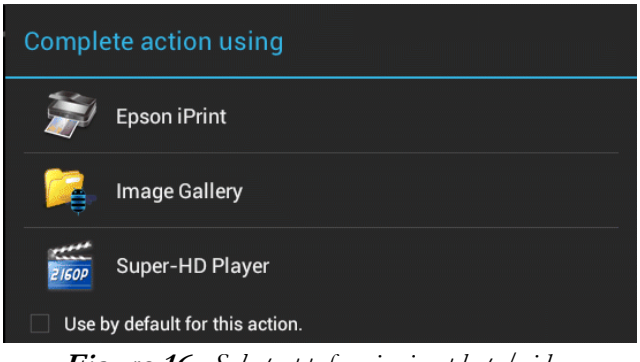


Figure 16 *- Select app for viewing photo/video.*

Figure 17 *- App displays photo/video.*

To exit the picture or video and return to the Checklist, press the Android "BACK" button .

3.3.4 Checks

A typical Checklist will include a number of steps for checking equipment features such as chassis, line cord, controls, indicators, displays and similar items. Checks have two forms: pass-fail inspection and pass-fail with optional user data input.

3.3.4.1 Pass-Fail Inspection

A pass-fail inspection will prompt you to check an item and enter the result by means of radio buttons. The selected result is recorded when the **'Next Step'** or **'Previous Step'** button is pressed.

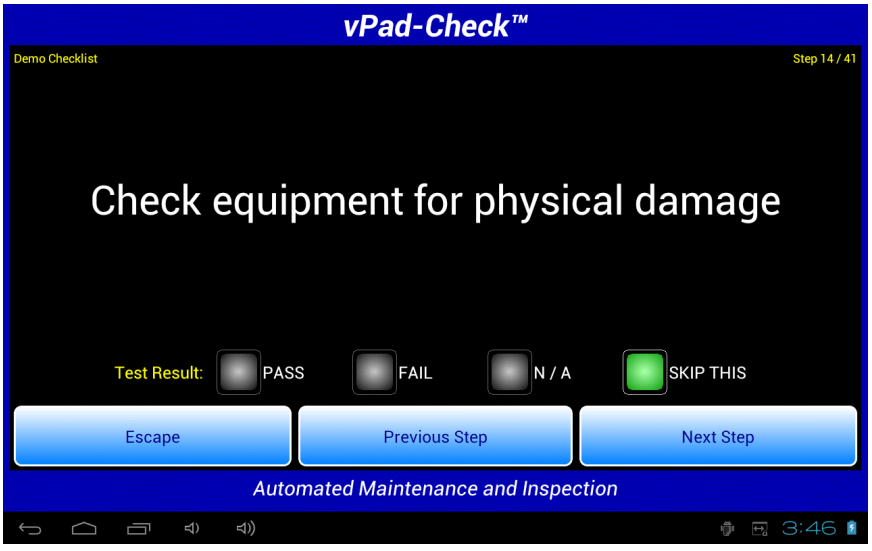


Figure 18 *- Pass-fail inspection.*

For a pass-fail inspection step, results which may be saved are:

PASS	The inspected item meets requirements or is acceptable.
FAIL	The inspected item does not meet requirements or is unacceptable. Note that if a "fail" is recorded for an inspection step, the Test Record will be assigned a "fail" for the overall test result.
N / A	This item is not applicable to the equipment being inspected or tested.
SKIP THIS	This item has not been inspected. This is the default result on entry to a pass-fail inspection step.

The **'Previous Step'** button may be used to step backward through a Checklist-in-progress, allowing you to review results you recorded for prior pass-fail inspection steps. If necessary, a result can be changed by pressing the appropriate button and the Checklist returned to the current step by the **'Next Step'** button.

3.3.4.2 Pass-Fail With User Data Input

This is similar to the pass-fail inspection described above, except that a "**Comments/Data**" box is provided to permit recording of additional information or test data. To enter a comment or data, tap the box and input the information using the Android keyboard. When done, dismiss the keyboard by pressing the button. For more details on using the Android keyboard, refer to Section 3.4.

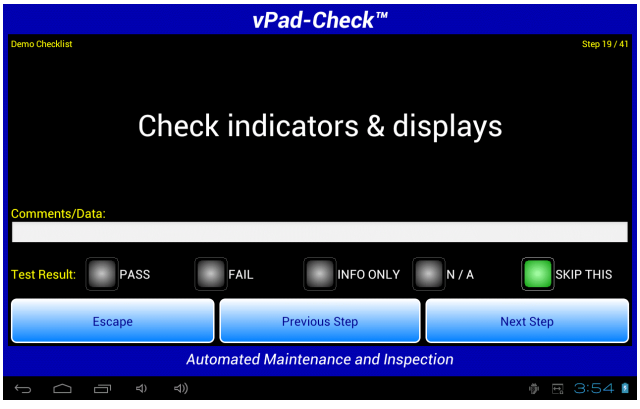


Figure 19 *- Pass-fail check with user input.*

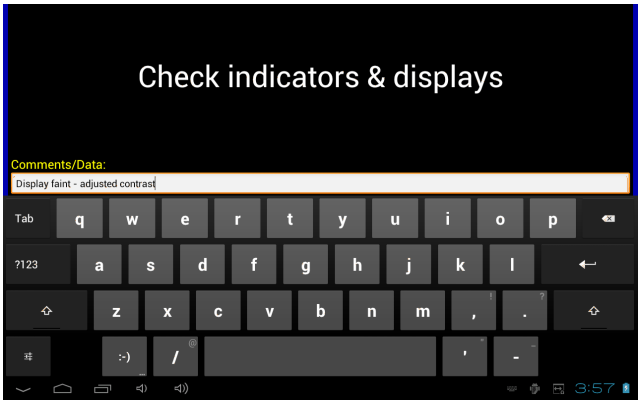


Figure 20 *- Entering inspection information.*

In addition to **'PASS'**, **'FAIL'** and **'N / A'** results described previously, pass-fail with user data input provides a button for classifying an inspection result as **'INFO ONLY'**. Select this result when recording information that has no bearing on equipment acceptability or performance. For example, you may select **'INFO ONLY'** when prompted to input general information such as dimensions, weight, software version and similar facts about the equipment under service.

3.3.5 Limit Tests

A limit test will require you to make a measurement and enter a numerical value, which is then compared to "pass/fail" limits that are in the Checklist. Limit tests are typically used for checking performance of the equipment under service, and for inspection steps where the inspected item is quantitative (e.g., operating temperature, fluid level, battery voltage, etc.)

When prompted by a limit test, make the measurement as instructed then tap the "**Test Data**" box to display the Android keyboard. Enter the measurement result using the keyboard, then press the "**Check Limits**" button.

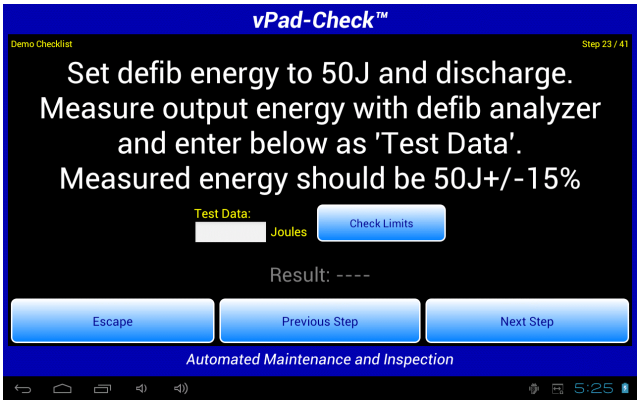


Figure 21 *- Limit test.*

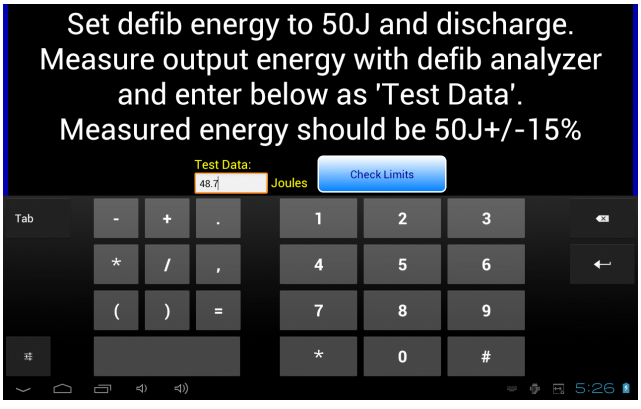


Figure 22 *- Entering a measurement value.*

The value you entered will be compared to the limits defined in the Checklist and a "PASS" or "FAIL" result will then be shown at the bottom of the screen. Press **'Next Step'** to save the result and continue with the procedure.

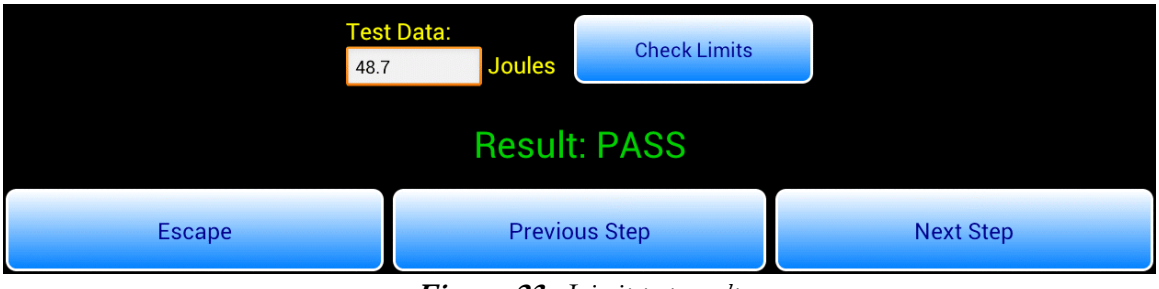


Figure 23 *- Limit test result.*

3.3.6 ECG Tests

If a Checklist is intended for a patient monitor or similar medical device, the Checklist may include steps for checking ECG performance. For these steps, the vPad-Check App will control the ECG simulator of the Analyzer Base Unit and you will be observing the corresponding response of the equipment under test. ECG test steps will typically be proceeded by 'prompt' steps providing instructions for connecting the vPad ECG simulator to the equipment.

When the vPad ECG simulator has been activated, the simulator waveform will appear on the screen. Observe the response of the equipment and enter the result requested by the Checklist. Depending on the Checklist, the requested result may be a simple pass/fail or a pass/fail with user data input. For ECG NSR simulations, a limit test may be shown. Some examples of ECG tests are shown below, which are from the "Demo Checklist" provided with vPad-Check. For additional instructions on operation of these menus, refer to Section 3.3.4 (Checks) or 3.3.5 (Limit Tests).

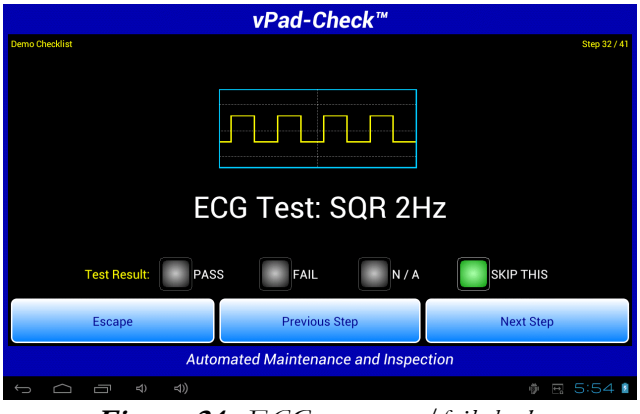


Figure 24 *- ECG test - pass/fail check.*

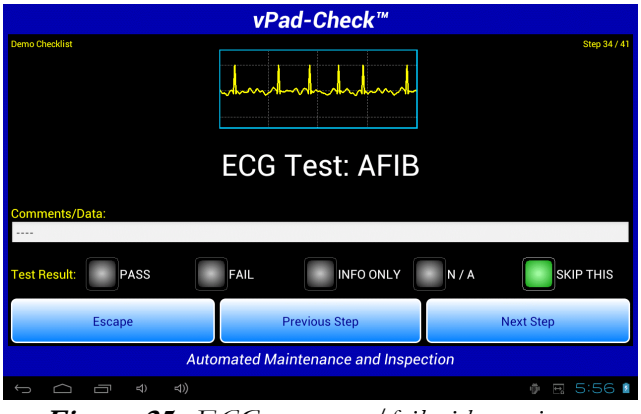


Figure 25 *- ECG test - pass/fail with user input.*

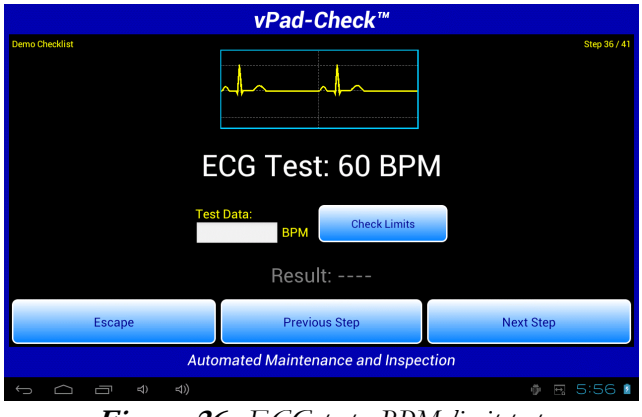


Figure 26 *- ECG test - BPM limit test.*

3.3.7 Electrical Safety Tests

A Checklist may include one or more electrical safety tests of the equipment under service. Checklists typically preface an electrical safety test with a prompt that informs you the test will be started at the next step.

Figure 27 *- Safety test prompt.*

When a safety test step is encountered, vPad-Check launches either the vPad-AS™ App or the vPad-A3™ App to perform the test. vPad-Check passes the AutoSequence number defined in the Checklist to the specified safety testing App. If **Enable Edits** has been checked in vPad-AS or vPad-A3, the AutoSequence Editor screen(s) will be enabled, otherwise, the **Editor** will be skipped and the sequence will begin by directly running the instructions in the sequence. If **Edits** have been enabled, the safety testing App will begin after a short delay, displaying the test settings for the selected AutoSequence.

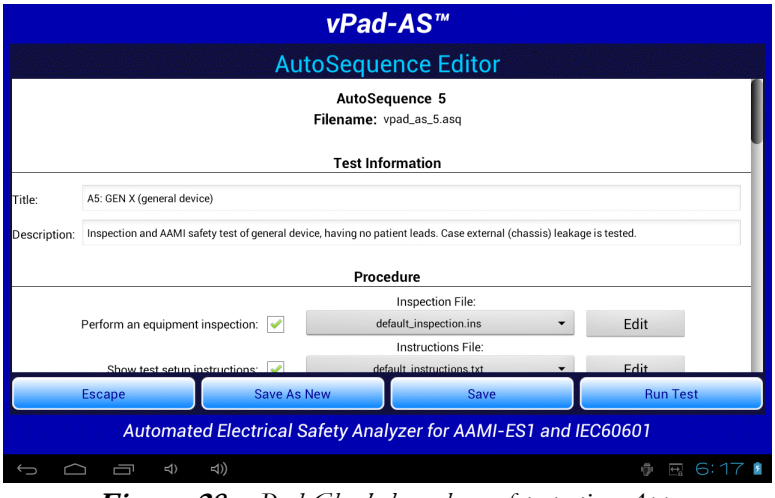


Figure 28 *- vPad-Check launches safety testing App.*

At this stage, you may review the test settings and make changes if necessary. Start the safety test by pressing the **'Run Test'** button.

If the safety test settings include an equipment inspection, this inspection is bypassed since all such inspections should be completed within the scope of the Checklist. Typically, the safety test sequence will begin by displaying instructions for connecting the equipment under test to the vPad Analyzer Base Unit. Press **'Next Step'** when ready to proceed with the test.

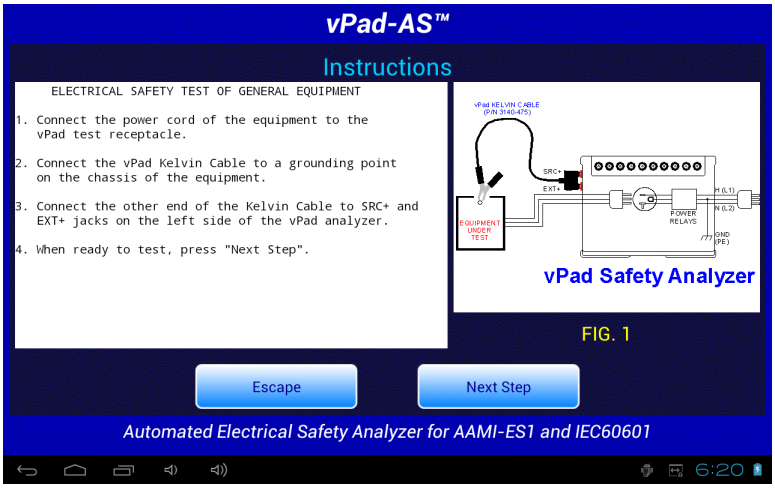

vPad-AS™
Instructions
ELECTRICAL SAFETY TEST OF GENERAL EQUIPMENT
1. Connect the power cord of the equipment to the vPad test receptacle.
2. Connect the vPad Kelvin Cable to a grounding point on the chassis of the equipment.
3. Connect the other end of the Kelvin Cable to SRC+ and EXT+ jacks on the left side of the vPad analyzer.
4. When ready to test, press "Next Step".
vPad Safety Analyzer
FIG. 1
Escape
Next Step
Automated Electrical Safety Analyzer for AAMI-ES1 and IEC60601
6:20

Figure 29 *- Safety test setup instructions.*

Following the instructions, the safety test will proceed automatically through a number of measurements. Each measurement is compared to a limit which is defined in the safety AutoSequence setup.

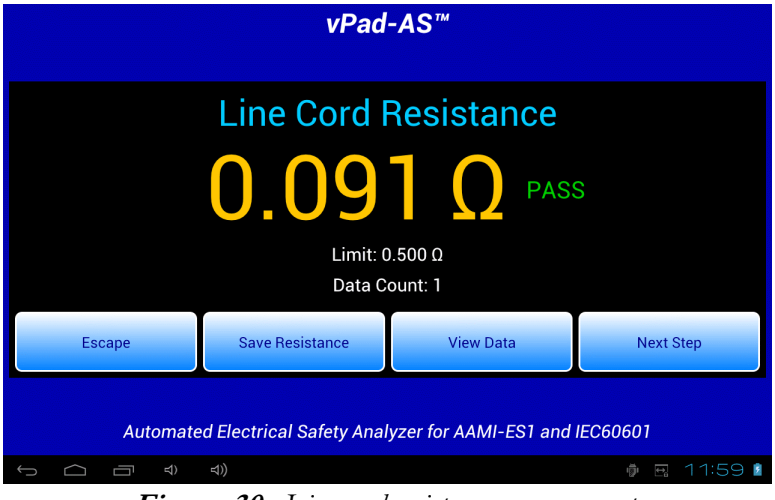


Figure 30 *- Line cord resistance measurement.*

The measurement sequence may stop at certain steps and you may be required to respond to prompts or alarms displayed by the safety testing App. If necessary, refer to **MN-079 (6100-045)** ***vPad-AS Operators Manual*** or **MN-080 (6100-046)** ***vPad-A3 Operators Manual*** for further information on electrical safety testing and use of these Apps. Note these operator manuals are provided on the Android tablet, and may be viewed with the **Datrend Docs App**

which is on the Home screen.

After all measurements of the safety test sequence have been completed, you are presented with a menu for saving the acquired results in the Test Record. The overall result of the electrical safety test is shown as "PASS" or "FAIL"; if the result is a "PASS" and you wish to change it to a "FAIL" you may do so by pressing the radio button. If the result is a "FAIL", you cannot modify the overall result. If desired, enter concluding comments about the safety test by tapping the **'Final comments'** box, which will display the Android keyboard. When ready to exit the safety testing App and return to the Checklist, press the **'Save Safety Test'** button.

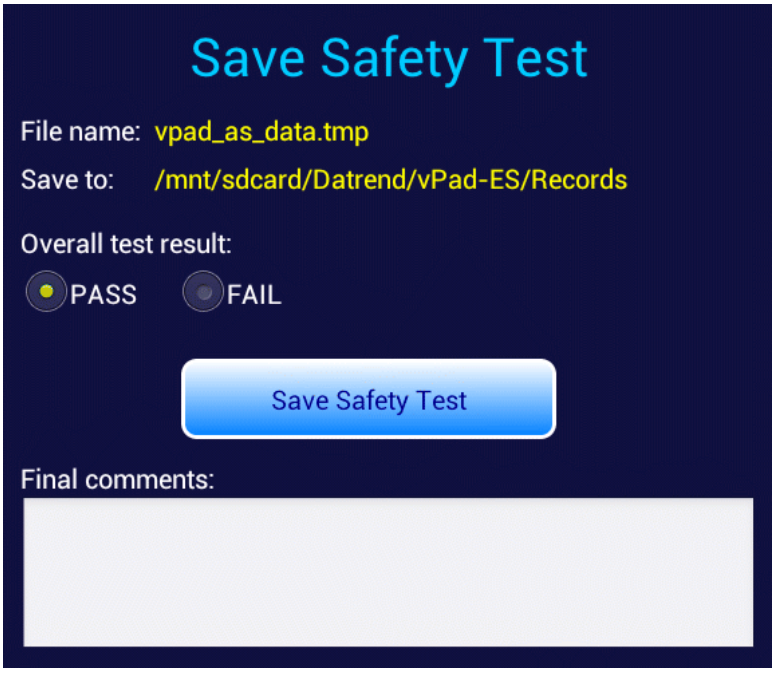


Figure 31 *- Save safety test results.*

After the safety testing App is closed, measurement values and the overall result of the safety test are then displayed by the Checklist. Results are displayed in a scrollable view; to see all values, move the view by press-and-drag gesture. Press **'Next Step'** when ready to proceed to the next step of the Checklist.

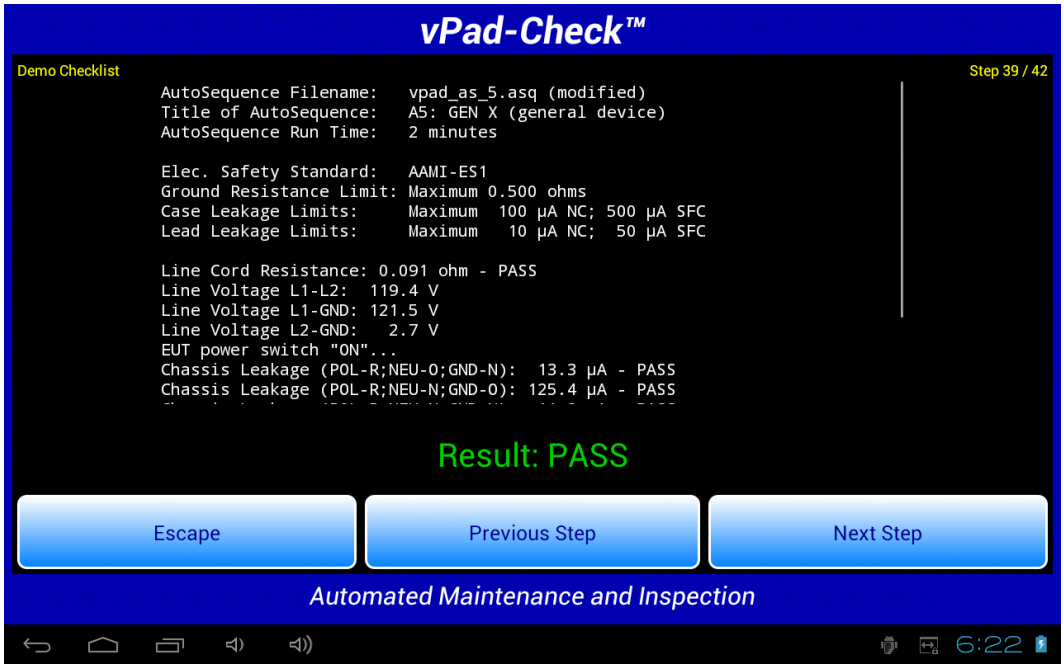

vPad-Check™
Demo Checklist
Step 39 / 42
AutoSequence Filename: vpad_as_5.asq (modified)
Title of AutoSequence: A5: GEN X (general device)
AutoSequence Run Time: 2 minutes
Elec. Safety Standard: AAMI-ES1
Ground Resistance Limit: Maximum 0.500 ohms
Case Leakage Limits: Maximum 100 µA NC; 500 µA SFC
Lead Leakage Limits: Maximum 10 µA NC; 50 µA SFC
Line Cord Resistance: 0.091 ohm - PASS
Line Voltage L1-L2: 119.4 V
Line Voltage L1-GND: 121.5 V
Line Voltage L2-GND: 2.7 V
EUT power switch "ON"...
Chassis Leakage (POL-R;NEU-O;GND-N): 13.3 µA - PASS
Chassis Leakage (POL-R;NEU-N;GND-O): 125.4 µA - PASS
Result: PASS
Escape
Previous Step
Next Step
Automated Maintenance and Inspection
6:22

Figure 32 *- Safety test results displayed by vPad-Check.*

While a safety test is running, you may press the **'Escape'** button to abort the test. The safety testing App will close and execution will return to vPad-Check. No electrical safety measurements are recorded in this case. Instead, vPad-Check will indicate the safety test has been cancelled, and this cancellation will also be indicated in the Test Record as a "skipped" safety test.

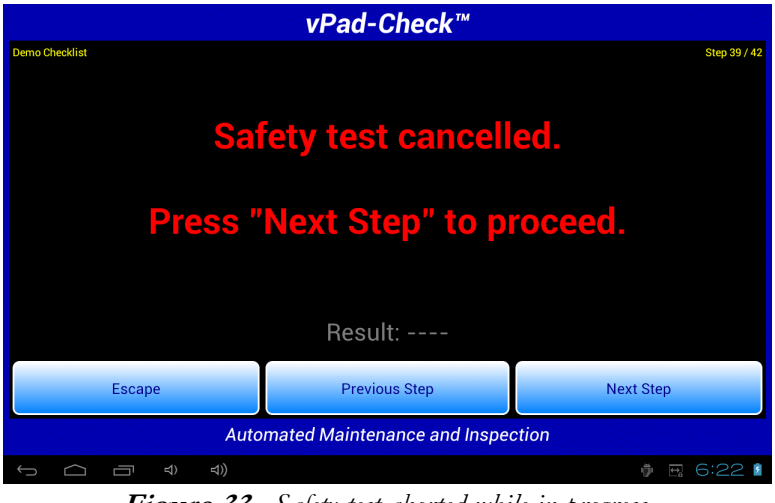


Figure 33 *- Safety test aborted while in-progress.*

If you abort a safety test, you can restart the safety test from the beginning by pressing the **'Previous Step'** button, or you can press **'Next Step'** to continue with the Checklist without completing the safety test. To abort the Checklist and discard all results, press **'Escape'**.

Most Checklists include an electrical safety test as described in this section. However, it is possible that a Checklist may have more than one safety test, in which case the safety testing App may be launched *several times* over the course of an inspection and maintenance procedure. Checklist procedures for critical medical devices may include two or more safety tests to ensure thorough conformance to a particular standard, for example, the IEC60601-1 standard which requires chassis leakage (touch current) tests as well as ground wire (earth) leakage tests. Electrical safety requirements for general devices and non-critical equipment are usually satisfied by a single safety test which has been specifically chosen to suit the equipment under service, and most example Checklists provided with vPad-Check are of this type.

Checklists can also launch other vPad Apps such as vPad-Cal™, which enables the user to include information about the test instruments used during the procedure.

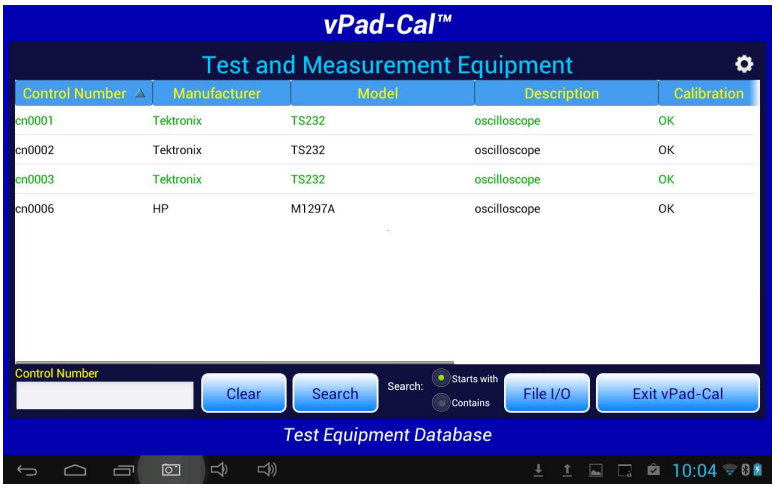


Figure 34 *- Selecting Test Instruments.*

When invoked by vPad-Check, vPad-Cal allows the user to select one or more test instruments for inclusion in the test record file. Selected instruments will appear with a green highlighting. When all required instruments have been selected press the Exit vPad-Cal button and choose the verbose or concise format for inclusion in the test record. Verbose includes all available data on the test instrument, concise provides a one line per instrument report, as shown below.

Refer to **MN-088 (6100-080)** ***vPad-Cal Operators Manual*** for further information on use of vPad-Cal.

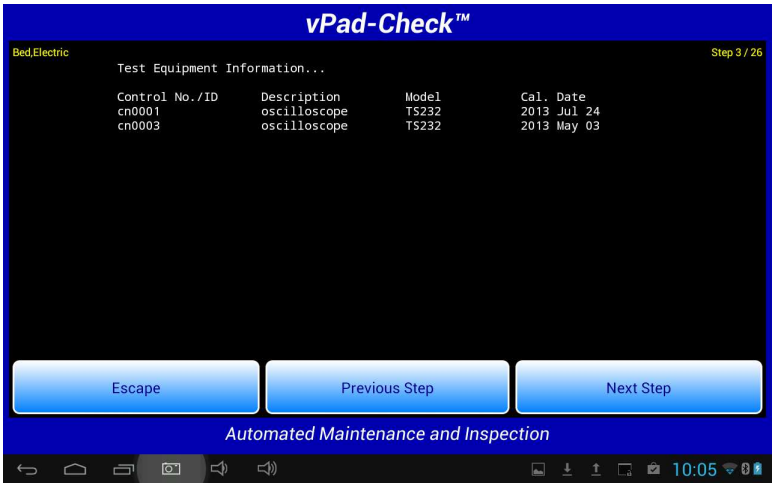


Figure 35 *- Test Instrument Record - Concise.*

3.3.8 Remote Control

A Checklist may include steps which automatically control an external test instrument such as a defibrillator analyzer or a patient simulator. Through remote control, a Checklist may send messages to a test instrument to configure it for a particular measurement or performance test, and may also acquire measurement values or test results from the instrument which may then be passed or failed. Test data acquired by remote control is combined with other results saved while performing an inspection and maintenance procedure.

This section explains how to set up a test instrument or patient simulator for use with vPad-Check, and provides examples of Checklist steps associated with automated remote control of such test equipment.

3.3.8.1 Remote Device Connection

To connect external test equipment to the vPad, an XBUS Smart Cable or "***XSC***" interface will be required. XSC adaptors are available for wired or wireless interface to test devices, with the wireless option being capable of both wired and wireless operation.

TABLE 3.3-1 lists Datrend part numbers for XSC available at the time of writing of this manual. Additional XSC interfaces are to be released for test products from Datrend systems Inc. as well as existing competitive and legacy test products from Fluke® Biomedical, DNI Nevada, Bio-Tek Instruments and others.

TABLE 3.2-1: XSC Interfaces for Test Equipment

XBUS Smart Cable	Wired Option (serial only)	Wireless Option (Bluetooth & serial)
Interface for Datrend AMPS-1, MPS-2 and FMS-3 multi-parameter simulators	7400-104	7400-103
MBOX universal interface	7400-106	7400-105

The "***MBOX***" (part no. 7400-105 or -106) is a universal adaptor which provides a basic interface to almost any test device or patient simulator that can be remotely controlled through a RS-232 serial port. With MBOX, a Checklist can send control messages to a remote test device and receive data which is sent by the device to vPad. MBOX provides data communication but results acquired from a test instrument through MBOX must be manually passed or failed by the user.

Figure 36 shows the general setup for a wired connection to a remote device, in this case a Datrend AMPS-1 patient simulator. The figure is a schematic representation and is not drawn to scale. General setup instructions for wired connections are as follows:

a. Using the RJ11-6 cable provided with the XSC interface, connect the "XBUS" Port of the vPad Base Unit to either of the "**XBUS**" inputs of the interface. Note that one of the XBUS inputs will remain available to permit daisy-chaining of additional XSC units for remote device interfacing.

b. Connect the RS-232 serial port of the test equipment to the "**DEVICE**" output of the XSC interface.

c. Apply power to the test equipment when required. The XSC adaptor receives its power from the vPad Base Unit, through the XBUS connection.

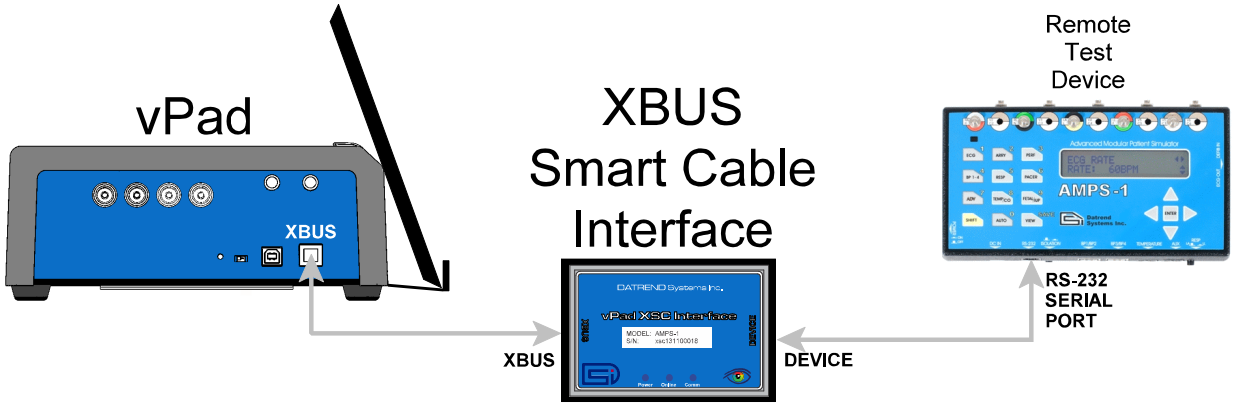


Figure 36 *- Schematic showing setup of remote test device - wired (serial) interface.*

Figure 37 shows the general setup for a wireless connection to a remote test device. The figure is a schematic representation and is not drawn to scale. Communication is based on point-to-point Bluetooth and is possible over distances of up to thirty feet (10 meters).

Wireless setup instructions are as follows:

a. Using the RJ11-6 cable provided with the XSC interface, connect the "XBUS" Port of the vPad Base Unit to either of the "**XBUS**" inputs of the interface. Note that one of the XBUS inputs will remain available to permit daisy-chaining of additional XSC units for external device interfacing.

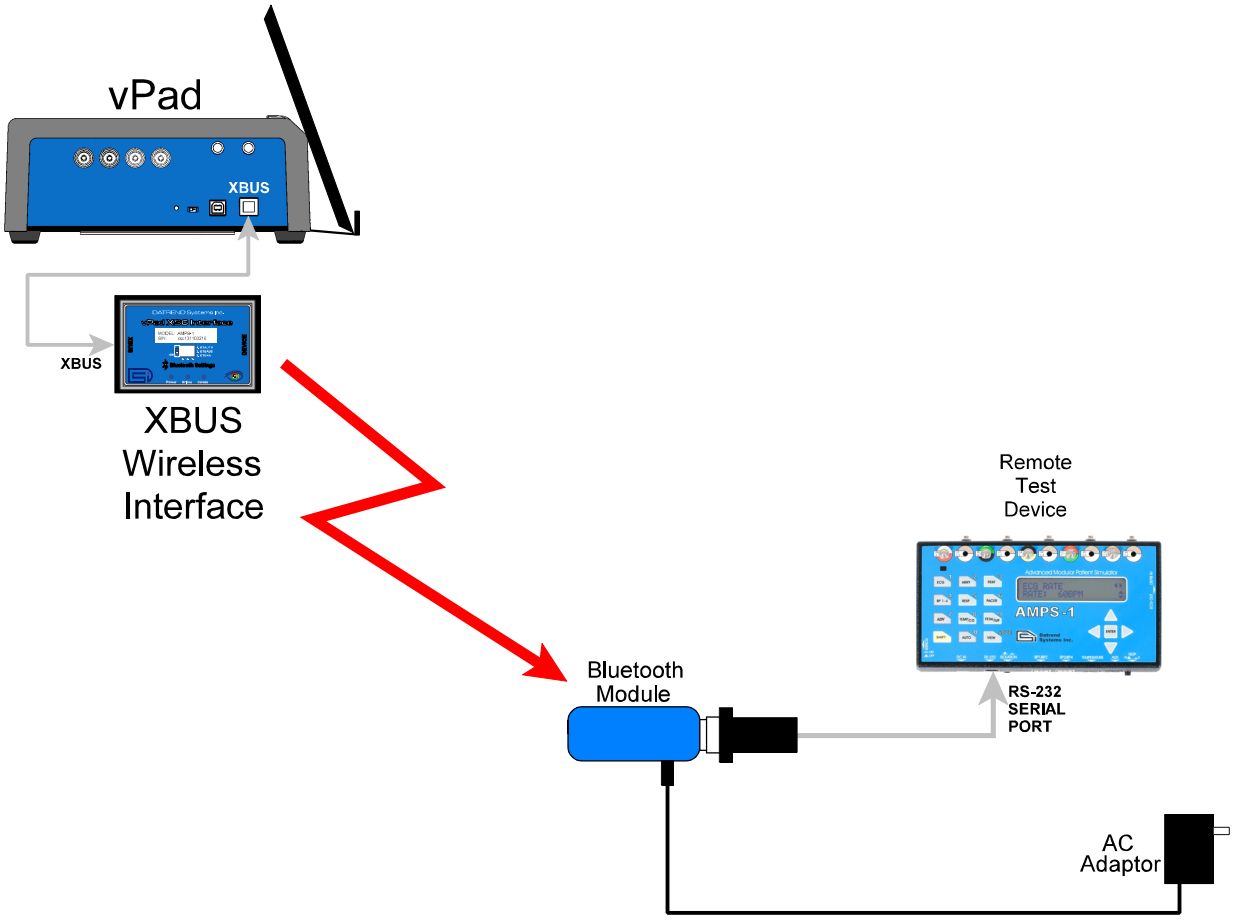


Figure 37 *- Schematic showing setup of remote test device - wireless (Bluetooth) interface.*

b. Connect the Bluetooth Module provided with the XSC interface to the serial port of the test equipment. Note this Bluetooth Module has been configured to communicate specifically with its companion XSC interface.

c. Connect the AC adaptor provided with the XSC interface to the DC power input of the Bluetooth Module, then connect the adaptor to the AC supply.

e. Apply power to the test equipment when required. The XSC interface receives its power from the vPad Base Unit, through the XBUS connection. When powered, the XSC interface will seek out the Bluetooth Module at the remote device, and if the Module is powered up, a point-to-point radio link will be established automatically.

f. Optionally, a wireless XSC may be switched to wired operation. To do so, disconnect the XSC from the vPad and set the "BTENA" switch of the interface to the "OFF" position. This will disable the Bluetooth connectivity of the XSC. A wired connection to the test equipment may then be set up as shown in ***Figure 37*** .

3.3.8.2 Remote Control Steps

A Checklist which includes automated control of a remote test instrument or patient simulator will begin with a step that "opens" a connection to the remote device. This step may be placed anywhere in a Checklist procedure, although it is usually done near the start of the procedure. Establishing communication with the remote device will typically require one or two seconds, during which time the buttons of vPad-Check will be disabled. If connection with the remote device is successful, the Checklist will advance to the next step automatically.

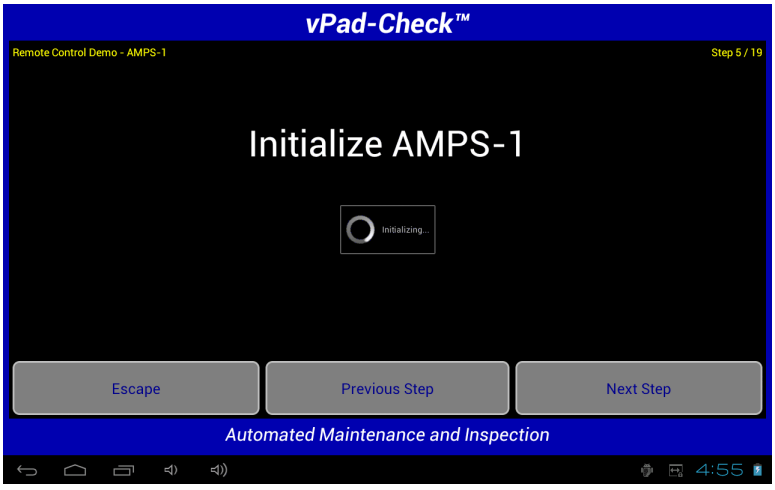


Figure 38 *- Initializing communication with remote test device.*

If the connection cannot be opened, an error message will be displayed. In this case, check physical connections per ***Figure 36*** or ***Figure 37*** and verify the test equipment is powered up. Note that certain test devices need to be set to "remote control mode" to enable serial communication; if necessary, refer to instructions provided with the test device. In the event of an communication error, you can correct the problem with the hardware and attempt to reconnect to the device by pressing **'Previous Step'** then **'Next Step'**.

A Checklist may include steps which send messages to a remote test instrument to configure the device or initiate a measurement procedure. When this type of step is executed, a description of the remote control command is displayed by vPad-Check while the Checklist delays a few seconds, as clocked by a timer which is indicated at the top of the screen. vPad-Check buttons are disabled during transmission of the control command to the remote device. After the delay, the Checklist proceeds to the next step automatically.

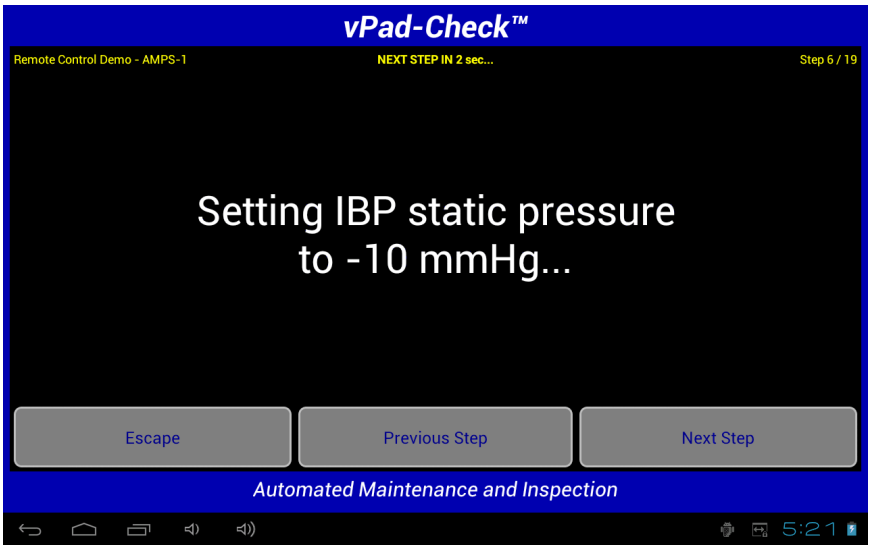


Figure 39 *- Sending a control command to a remote test device.*

A Checklist may include steps which poll a remote test instrument for measurement data or test results. Data from a test instrument is received and, depending on the test device and XSC adaptor in use, the data may be automatically assigned a pass/fail result or, the user may need to select "PASS" or "FAIL" manually. For received data which is automatically assigned a "PASS" by vPad-Check, the Checklist will advance to the next step automatically. For data which is not automatically assessed, select the appropriate result with the **'PASS'** or **'FAIL'** button and press **'Next Step'** to continue.

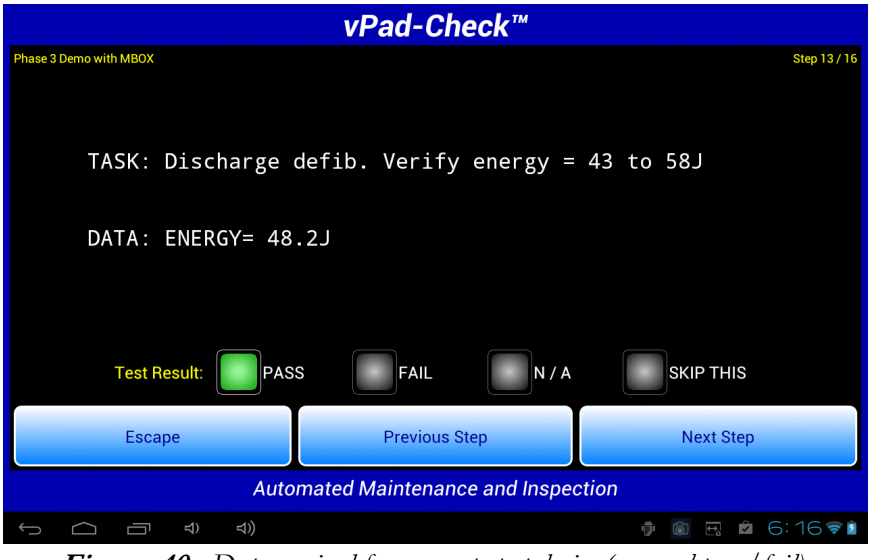


Figure 40 *- Data received from remote test device (manual pass/fail).*

3.3.9 Checklist Results Preview

When all steps of the Checklist have been completed, a preview of the results is then displayed. Tasks performed and data recorded during the procedure will be displayed in a ScrollView, as shown in the example below.

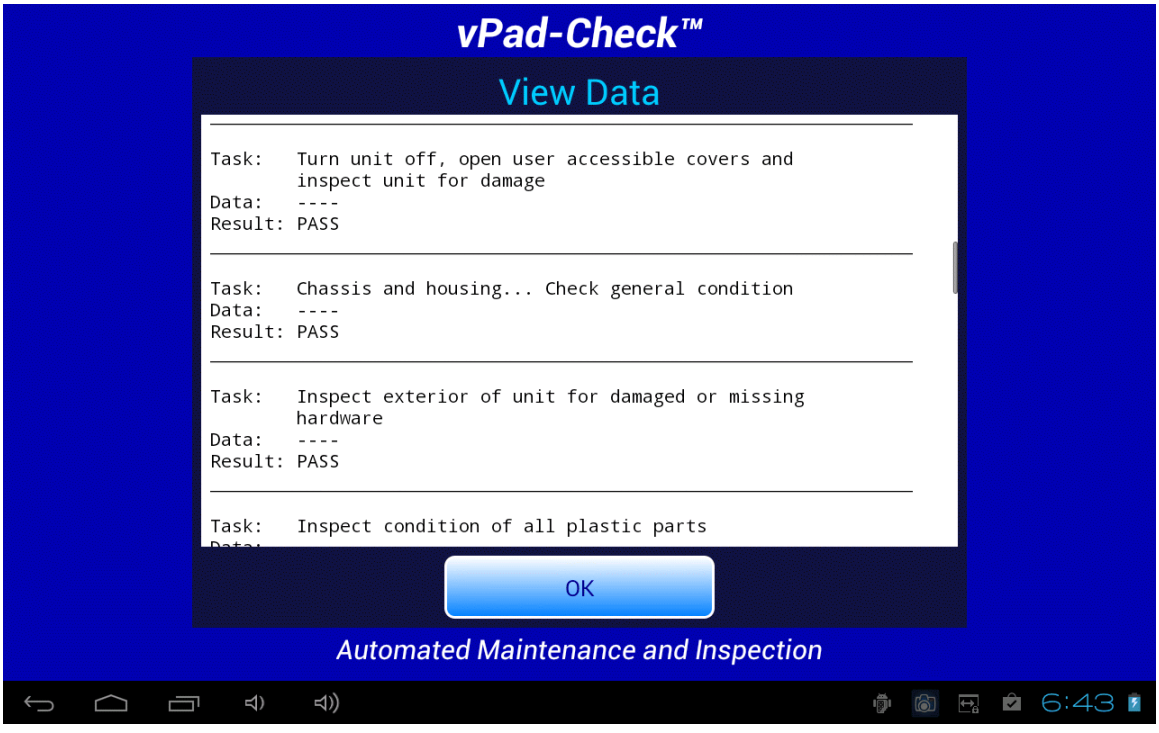


Figure 41 *- Checklist results preview.*

If necessary, the window can be scrolled up or down using a up/down touch-and-drag gesture. When done, press **'OK'** to proceed to the Create Record menu.

3.3.10 Creating a Test Record

When all results have been collected, the final step is to create a record of the test. Reports are stored in text files which have a name beginning with the EUT Control ID. The remainder of the file name includes a date and time stamp. The general format of the file name is:

<control-id>_<yyyymmdd>_<hhmmss>_<r>.txt

Where **<r>** can be a **F** for a failed test or **P** for a passed test.

In the event a Control ID has not yet been entered, a reminder warning will be displayed, as shown in ***Figure 42***. Pressing **'Continue'** will return to the Equipment Information screen (see ***Figure 9***, page 17). Otherwise, the **'Create Test Record'** screen, ***Figure 43***, will be displayed.

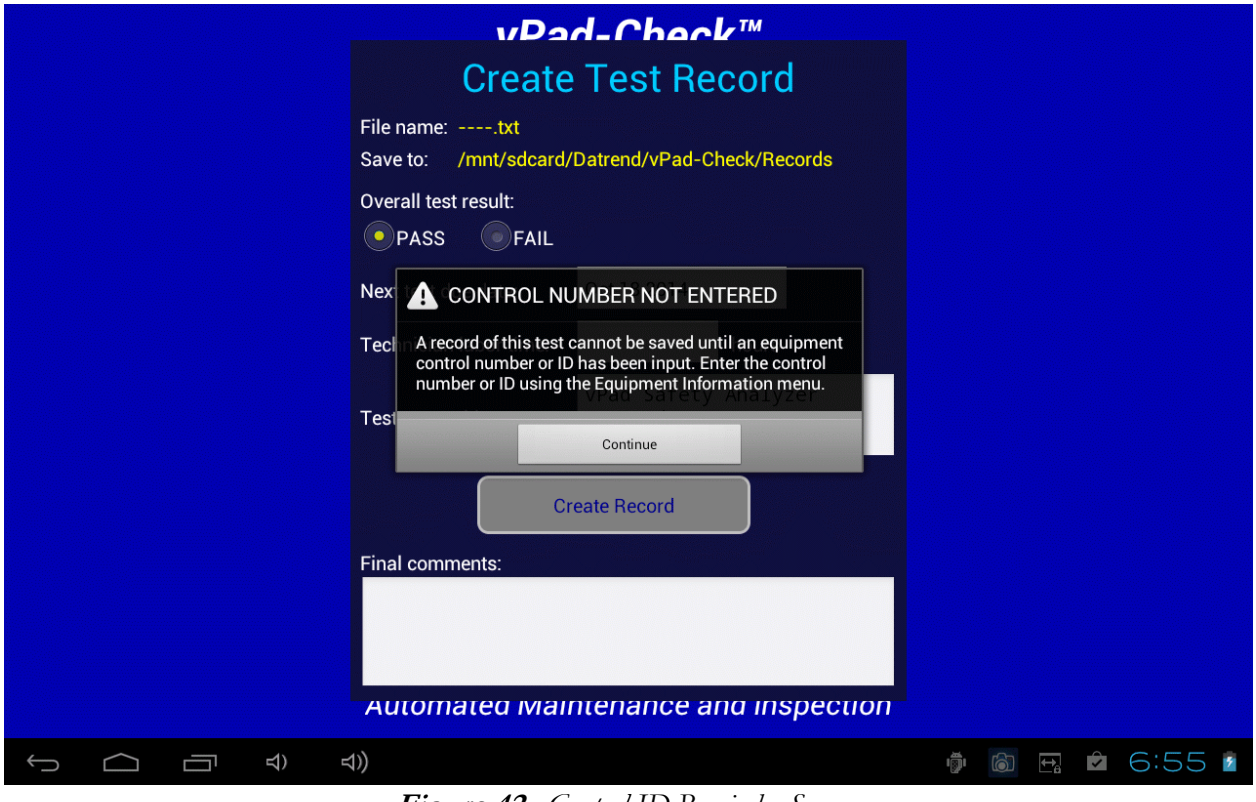

Create Test Record
File name: ----.txt
Save to: /mnt/sdcard/Datrend/vPad-Check/Records
Overall test result:
PASS
FAIL
CONTROL NUMBER NOT ENTERED
A record of this test cannot be saved until an equipment control number or ID has been input. Enter the control number or ID using the Equipment Information menu.
Continue
Create Record
Final comments:
6:55

Figure 42 - *Control ID Reminder Screen.*

Once the EUT has been identified by Control Number, it is possible to complete the creation of the Test Record (***Figure 43***). The filename and destination are displayed at the top of the screen and are not user editable.

Figure 43 - *Create a Record.*

During the test, certain measurements may have been recorded as a **FAIL** result. If this is true, the **'Overall Test Result:'** will activate the **FAIL** radio button. If there were no failures recorded during the test, the **PASS** radio button will be activated. It is possible to override a **PASS** since there may be circumstances not evident from the test measurements which could recommend a **FAIL** assessment. Reasons for changing a **PASS** to a **FAIL** can be recorded in the **'Comments:'** field.

Note: if more than 60% of the (Pass/Fail) steps in the procedure have been skipped, the procedure will be recorded as **FAIL**. This is an indication that the procedure needs to be revised or a more relevant procedure needs to be created for this type of device.

The **'Next test due date'** field shown on the Create Record menu (***Figure 43***) is the date at which the equipment becomes due for its *next* inspection or service. If a due date was entered by the user during input of the equipment information, then this date will be shown in the **'Next test due date'** field. If no due date was entered with the equipment information, then a default due date is automatically generated based on the current date plus the "next test" interval from the Label Printer Settings menu (see Section 3.2.3.2, page 14). If necessary, the due date shown may be adjusted by touching the field space which will

display the Android keyboard (***Figure 44***). The **'Next test due date'** is saved in the Test Record, and will appear on the PM inspection label if a label is printed.

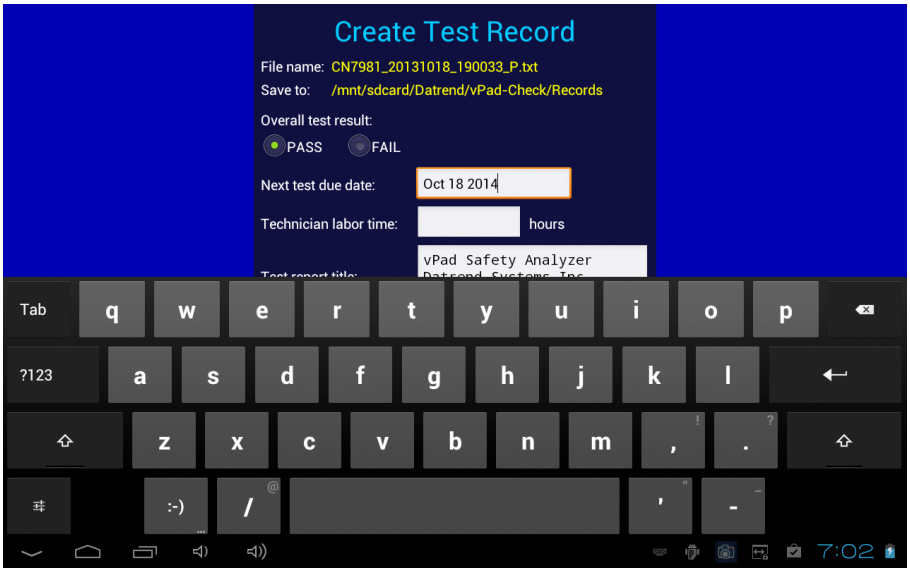


Figure 44 - *Modifying the default "next test due" date.*

Time spent by the technician/user can be recorded by entering a value in the **'Technician labor time:'** field. Touching this field space will result in the display of a numeric keypad to allow data entry, per ***Figure 45***. Note that this time is intended to encompass ALL of the time spent by the user in preparing for and completing the requirements of the work order. The actual time spent using vPad-Check to take and record measurements is recorded separately in the Test Record under the heading 'Checklist Run Time'. This is the time from the initial selection of the Checklist until the Test Record is created.

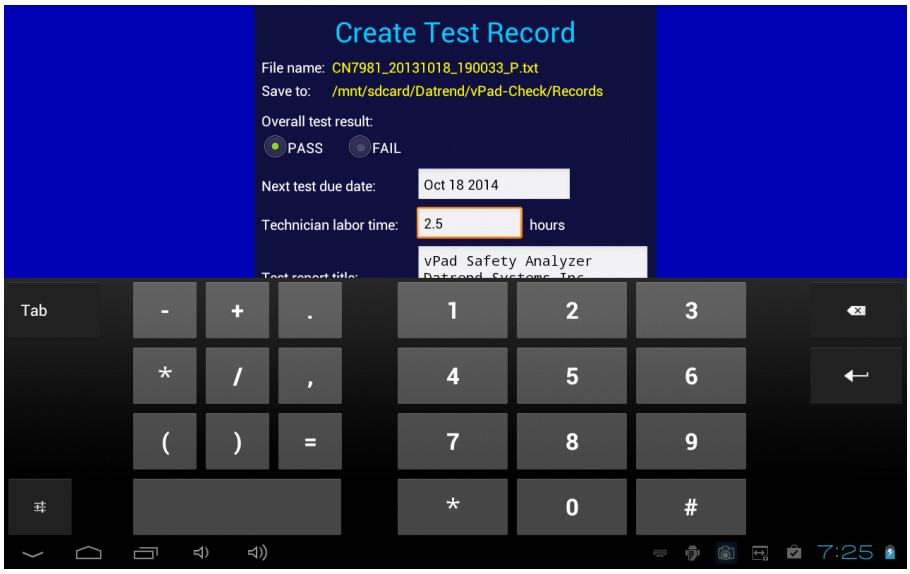


Figure 45 *- Entering Technician Time.*

Test Record files include a Title at the beginning of the file. This may be modified from default "**vPad Safety Analyzer**" after tapping the **Test report title** field. Optionally, enter the name of your institution or business, address, phone number and so on. Up to 16 lines may be entered into this text field, of which the first three lines will be shown when the menu is initially displayed (to view or edit additional lines, scroll the window using a press-and-drag gesture). The **Test report title** is backed up in permanent memory storage on the Android tablet; the title is maintained after power to the vPad is switched off, and is then restored on subsequent use of the app.

In the event there is other relevant information concerning the test, the 'Final Comments' field provides a means of recording the information. This could be reasons that a test result was changed from PASS to FAIL, notes regarding additional work that should be done in the future, requirements to order parts, etc.

Once all of the information has been input, press the **'Create Record'** button to save the Test Record.

3.3.11 Printing an Inspection Label

If label printing has been enabled, after a Test Record is saved a **'Print Label'** menu will then be shown (***Figure 46***).

Figure 46 *- Print PM inspection label.*

This menu displays information as it will be printed on the inspection label, for example, the equipment ID, test date and the next test due date. Refer to Section 3.2.3 for printer setup instructions. When the printer is ready to print, press the **'Print Label'** button. Multiple labels may be printed if necessary. When done, press **'Exit'** to return to the main menu of vPad-Check.

3.4 Using the Android Keyboard

When you press a text field, the Android keyboard is displayed at the bottom of the screen.

Figure 47 *- Android onscreen keyboard.*

The onscreen keyboard is the default method used by Android to enter information. If you have connected an alternative input device to the tablet (e.g. Bluetooth/USB keyboard or barcode scanner), then the Android keyboard will not appear when a text field is pressed and data entry will then be from the external input device. To switch back to the default keyboard, disconnect the external input device.

The text field which is currently accepting data from the Android keyboard will be highlighted:

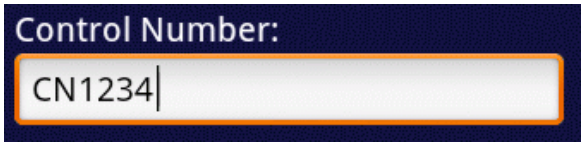


Figure 48 *- Single-line text field.*

Android text fields may be either single- or multi-line. A practically unlimited amount of text may be typed for either single- or multi-line fields.

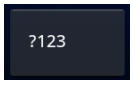
Pressing ENTER (or **Next)** for a single-line text field causes data entry to jump to the next text field on the screen.

Pressing ENTER on a multi-line text field inserts a new line. To move from a multi-line field to a different text field on the screen, tap on the desired field.

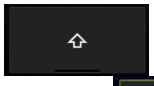
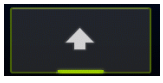
Press

to navigate forward through multiple text fields on a screen. Press

to navigate backward. You can also jump to a particular field on a screen simply by tapping the field.

Switch the Android keyboard from alpha to numeric modes by pressing 

.

In alpha mode, to activate Caps Lock press and hold the Shift key  or double-tap the Shift key so that the indicator lights up:  . Once activated, you can press to switch Caps Lock off.

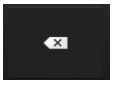
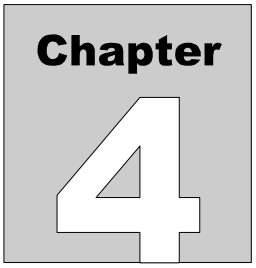
To correct a mistake in typing, tap the text field to the right of the error. This will activate the Android edit cursor. Erase with backspace/correct  then retype.

Figure 49 *- Error correction with edit cursor.*

When done entering text, the Android keyboard may be removed from the screen by pressing at bottom left.



4 vPad Programming Language

vPad "Checklists" correspond to a set of files which are stored in the vPad-Check directory of the Android tablet. A Checklist file contains a ***computer program*** which has been written in the vPad Programming Language or "***vPL***". Each step of a Checklist procedure has a direct one-to-one relationship with a corresponding vPL ***statement*** in the Checklist file. vPL statements can display instructions, show photographs and video, accept data input from the user, run electrical safety tests, communicate with external test instruments and perform other functions.

This chapter specifies the vPad Programming Language and explains how vPL may be used to create Checklist procedures.

4.1 Checklist Files

4.1.1 File Location

Checklist files are stored in this directory of the Android tablet:

`<storage_media>/Datrend/vPad-Check`

The **`<storage_media>`** refers to the memory installed in the Android tablet and its name will depend on the make and model of the tablet. For most tablets, this is typically **`mnt/sdcard`**.

4.1.2 Filenames

Checklist files have a "**`.vpl`**" file extension. The filename will usually refer to the medical equipment under service, for example:

```
Defibrillator.vpl
Electrocardiograph.vpl
Incubator.vpl
Ventilator.vpl
```

Filenames may refer to medical equipment generically, as above, or they may refer to specific device manufacturers and model numbers.

A filename may also describe a particular type of maintenance procedure, for example:

```
How to Replace Zoll AED Battery.vpl
```

As in the above example, filenames may include whitespace characters and they can also include comma characters. Here are a few more examples showing use of whitespace and commas in filenames.

```
Monitor, CO2.vpl
Monitor, ECG.vpl
Monitor, Patient, Multi-parameter.vpl
Monitor, NIBP.vpl
Monitor, SpO2.vpl
Pump, Enteral Feeding.vpl
Pump, IV Infusion.vpl
Pump, Suction.vpl
```

Since the Checklist selector menu of vPad-Check displays the Checklist files alphabetically, the above is a useful way of naming Checklists which will make it easier for a user to search for a procedure of interest.

4.1.3 File Editing

Checklist files are text files that may be edited with any text editor or word processor.

You can use the text editor app installed on the Android tablet to edit Checklist files. Alternatively, you can copy vPad-Check file(s) to a personal computer (PC), make changes, and copy the file(s) back to the tablet. vPad-Check provides an "Update Files" feature to help with transferring files from the PC to the tablet (see Section 3.2.2).

You may find it easier to write Checklists on a PC with a ***source code editor***. A source code editor is a text editor program designed specifically for editing source code of computer programs by programmers. Source code editors provide ***syntax highlighting*** which makes computer programs easier to read.

A source code editor that allows you configure the syntax highlighting is best for programming Checklists. EditPlus (www.editplus.com) is an example of a source code editor with user-definable syntax settings. EditPlus is shareware, but there are other editor programs which can be downloaded from the internet for free from various sources (e.g. http://sourceforge.net/).

Figure 51 shows how a vPad Checklist appears when it is being edited with EditPlus[2]. Note how the source code editor has automatically highlighted various items within the vPL program code (green for comments; blue for keywords; grey for text; and red for arguments).

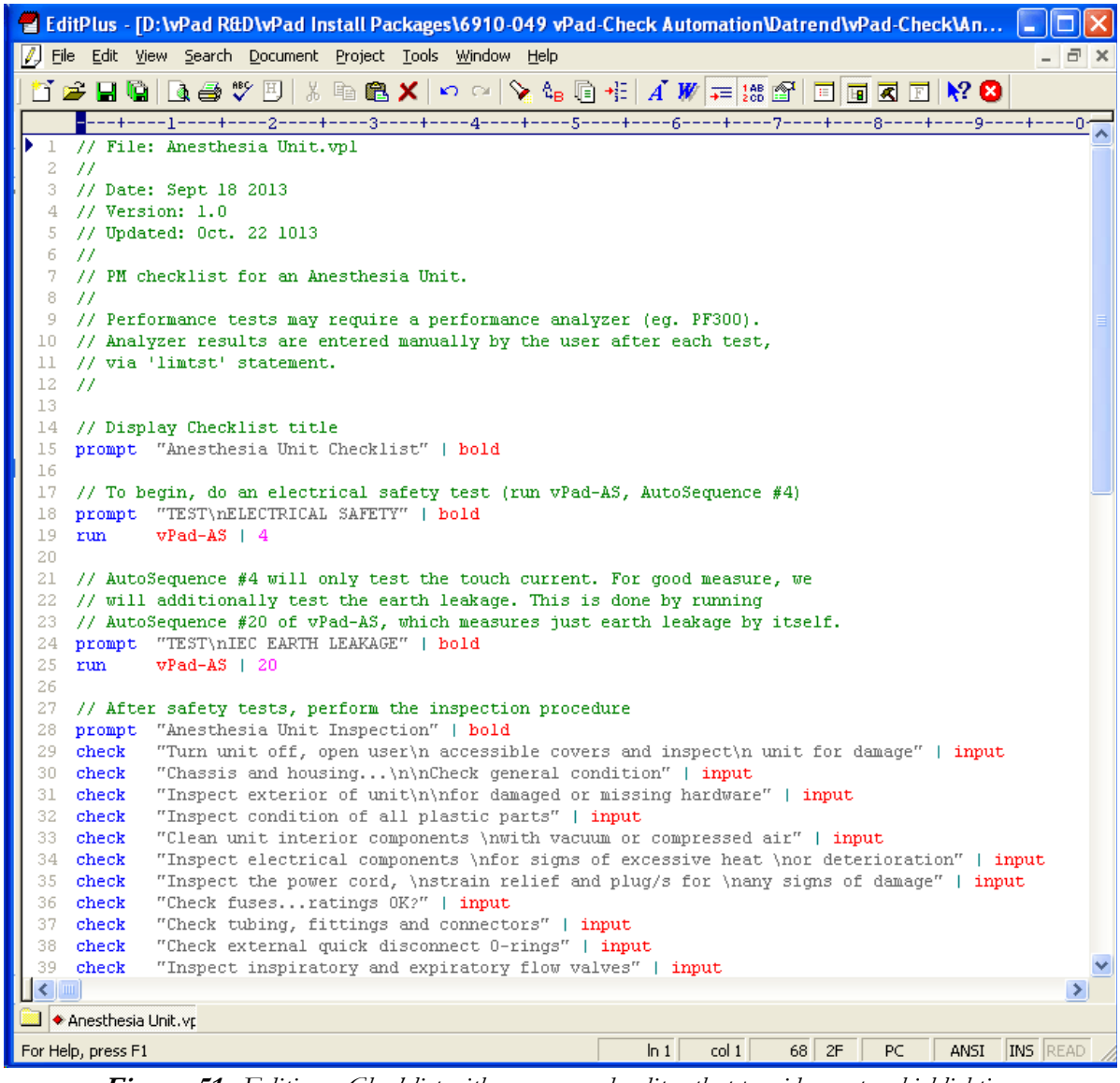

EditPlus - [D:\vPad R&D\vPad Install Packages\6910-049 vPad-Check Automation\Datrend\vPad-Check\An...
File Edit View Search Document Project Tools Window Help
// File: Anesthesia Unit.vpl
//
// Date: Sept 18 2013
// Version: 1.0
// Updated: Oct. 22 1013
//
// PM checklist for an Anesthesia Unit.
//
// Performance tests may require a performance analyzer (eg. PF300).
// Analyzer results are entered manually by the user after each test,
// via 'limtst' statement.
//
// Display Checklist title
prompt "Anesthesia Unit Checklist" | bold
// To begin, do an electrical safety test (run vPad-AS, AutoSequence #4)
prompt "TEST\nELECTRICAL SAFETY" | bold
run vPad-AS | 4
// AutoSequence #4 will only test the touch current. For good measure, we
// will additionally test the earth leakage. This is done by running
// AutoSequence #20 of vPad-AS, which measures just earth leakage by itself.
prompt "TEST\nIEC EARTH LEAKAGE" | bold
run vPad-AS | 20
// After safety tests, perform the inspection procedure
prompt "Anesthesia Unit Inspection" | bold
check "Turn unit off, open user\n accessible covers and inspect\n unit for damage" | input
check "Chassis and housing...\n\nCheck general condition" | input
check "Inspect exterior of unit\n\nfor damaged or missing hardware" | input
check "Inspect condition of all plastic parts" | input
check "Clean unit interior components \nwith vacuum or compressed air" | input
check "Inspect electrical components \nfor signs of excessive heat \nor deterioration" | input
check "Inspect the power cord, \nstrain relief and plug/s for \nany signs of damage" | input
check "Check fuses...ratings OK?" | input
check "Check tubing, fittings and connectors" | input
check "Check external quick disconnect O-rings" | input
check "Inspect inspiratory and expiratory flow valves" | input
Anesthesia Unit.vp
For Help, press F1
ln 1 col 1 68 2F PC ANSI INS READ

Figure 51 - *Editing a Checklist with a source code editor that provides syntax highlighting.*

[2] Requires installation of **`vpl.stx`** syntax file for EditPlus, available from Datrend Systems. See contact information in Chapter 5.

4.2 Checklist Structure

4.2.1 Blank Lines

Blank lines may be placed anywhere in a Checklist program and will be ignored by vPad-Check when the Checklist is executed. Blank lines are useful for separating a Checklist into logical sections.

4.2.2 Comments

Comments may be placed within a Checklist program to explain what the program does or how it works. Any line that begins with a double-slash **//** is considered a comment by vPad-Check and will be ignored when the Checklist is executed. A comment must be placed on its own line; a comment can not be placed inside of a vPL statement or on the same line as a statement.

```
// This is an example of a comment
// And here is another comment following the first comment
```

4.2.3 Statements

Each step of a Checklist procedure has a direct one-to-one relationship with a corresponding vPL ***statement*** in the Checklist.

A vPL statement begins with a ***keyword***. The keyword is then followed by a list of one or more ***arguments***, where the arguments are separated by the pipe character (**|**):

```
keyword    <argument_1> | [argument_2] | [argument_3] ...
```
[3]

There must be at least one whitespace between the keyword and the first argument. Arguments following the first may be optional and will depend on the keyword. The use of whitespace on either side of a pipe character is optional but helps to make the program more readable.

[3] In this manual, **<** **>** brackets are used to indicate the required or mandatory arguments of a vPL statement, and **[** **]** brackets are used to indicate optional arguments.

4.2.4 Keywords

The following keywords are defined in the vPL language:

Keyword	Application
`prompt`	Displays information or instructions to the user
`show`	Displays a drawing (PNG), photo (JPG) or video (MP4 or 3GP)
`check`	Pass/fail inspection, or pass/fail inspection with optional user input
`limtst`	Limit test; for quantitative inspections or performance tests where measurements are evaluated against set limits
`color`	Sets the foreground colour of text as displayed by **`prompt`**, **`show`**, **`check`** and **`limtst`** statements
`power`	Switches the vPad test receptacle ON or OFF
`ecgsim`	Controls the vPad ECG simulator
`run`	Runs an electrical safety AutoSequence (via vPad-AS or vPad-A3 App)
`xopen`	Opens a connection to a remote test instrument or a patient simulator via an XBUS Smart Cable interface (XSC)
`xctrl`	Controls a remote test device through a XSC interface
`xtest`	Runs a test or measurement procedure with a remote test device via an XSC interface

Keywords may be written in either upper or lower case letters. The use of lower case is a convention.

4.2.5 Strings

Following the keyword, the first argument of a vPL statement may be a ***string*** of characters. A string is a message that will be displayed by vPad-Check when the statement is executed.

A string may be surrounded by double-quotation marks. Enclosing strings with double-quotes is optional in vPL, but it is a convention to do so in other computer languages and it will make the strings appear highlighted in a different colour when using a source code editor (see ***Figure 51***).

If double-quotes are used inside of a string, then they will be removed before the message is displayed by vPad-Check. If it is necessary to emphasize a part of a message, use the single-quote character in strings.

Here are some examples of simple, single-line strings:

```
"Check fuses... ratings OK?"

"Inspect inspiratory and expiratory flow valves"

"Set switch to 'OFF' position"
```

In the last example, note the use of single-quotes to emphasize 'OFF'.

4.2.6 Newline Character Sequence

A line break may be placed inside of a string by using the **\n** character sequence. This is done to achieve a certain appearance or arrangement of the text when the message is displayed by vPad-Check.

Here is an example of a string which includes some newline characters:

```
"Fan(s) and filter(s)...\n\nCheck condition,\nclean and lubricate as required"
```

By default, vPad-Check will center-justify strings for display, unless a style is specified which overrides the default. If the above string was displayed with default center-justification, the three newlines **\n** will cause the message to appear as follows:

Fan(s) and filter(s)...

Check condition,
clean and lubricate as required

4.2.7 Line Continuation Sequence

Many statements will be short enough to be placed on a single line in the text editor. Some statements, however, may have a very long string, a large number of arguments, or both.

When a statement is too long to easily fit on a single line on the editor's screen, the statement may be broken up into several lines using the **\+** character sequence, which is placed at the end of each partial line of the statement. The **\+** sequence is used only to make the Checklist program more readable and it has no effect on how the statement will be executed by vPad-Check.

When a statement has a long string that has been formatted into paragraphs by the **\n** sequence, the **\+** sequence may be placed at the end of each line in the paragraph to make the statement more readable. Note it is the **\n** which controls the formatting of the displayed message, not the **\+**.

Here is an example of a **limtst** statement, illustrating the use of the **\+** sequence:

```
limtst  "Set defib to 50J & discharge.\n"\+
        "Measure output energy with defib analyzer\n"\+
        "and enter as 'Test Data'.\n"\+
        "Measured energy should be 50J+/-15%"\+
        | Joules \+
        | target \+
        | 50 \+
        | 15 \+
        | percent
```

This example shows a **\+** continuation sequence after each line of the string argument, and also following each additional argument after the string.

This statement can be rewritten with fewer line continuation marks. The following statement will produce exactly the same result from vPad-Check as the one shown above.

```
limtst "Set defib to 50J & discharge.\nMeasure output energy with defib analyzer\n"\+
       "and enter as 'Test Data'.\nMeasured energy should be 50J+/-15%" | Joules \+
       | target | 50 | 15 | percent
```

The statement could also be written with no **\+** sequences at all. In that case, the statement would likely extend outside the margins of a page as displayed by a text editor on a PC, but it would still be an executable statement as far as vPad-Check is concerned.

4.3 prompt Statement

Syntax: **`prompt <string> | [style] | [timeout]`**

UI Reference: Section 3.3.2

Description: The **`prompt`** statement displays a text string in a specified style. If the optional timeout is specified, the Checklist will automatically proceed to the next step when the timeout expires.

The string is the only required argument. The style is optional and if not specified a default style of **`normal`** will be used. When included, the optional timeout is specified as an integer number of seconds (1, 2, 3...). If a timeout is specified, then the style must also be specified.

Styles are defined in the table below. All styles will automatically word wrap to make the string fit in the designated space on the LCD display. The newline character sequence **`\n`** may be used in strings to override the default word wrapping.

Style	Font			Paragraph Justification	Left/Right Margins
	Rel. Size	Colour	Character Spacing		
`normal` (default)	60	white*	Proportional	Centered	0 pixels
`bold`	60-bold	white*	Proportional	Centered	0 pixels
`red`	60-bold	red	Proportional	Centered	0 pixels
`medium`	38	white*	Proportional	Left	0 pixels
`mmono`	32	white*	Uniform	Left	90 pixels
`small`	22	white*	Uniform	Left	200 pixels

* Default font colour for these styles is white. Text colour can be modified with the '**`color`**' statement. See Section 4.7.

Additional styles are:

bell This has the same appearance as **normal** style, and adds a bell sound which is played when the prompt is shown.

alert This has the same appearance as **red** style, and adds an alert tone which is played when the prompt is shown.

Examples:

```
prompt "Review manufacturer's instructions"


prompt "BEGIN EQUIPMENT INSPECTION" | bold


prompt "WARNING: HIGH VOLTAGE!" | red


prompt "ATTENTION!\nDisconnect hose from PORT A now." | bell


prompt "The equipment must reach operating temperature before "\+
       "testing. Switch the power 'ON' and allow about 15 "\+
       "minutes for the temperature to stabilize. The green "\+
       "indicator lamp on the equipment will be lit once the "\+
       "operating temperature has been reached.\n\n"\+
       "Press 'Next Step' when the equipment is ready..." | medium


prompt "UPSTREAM (Proximal) OCCLUSION SENSOR TEST:\n\n"\+
       "1. Press the ON/OFF key to turn the IV pump ON. Select\n"\+
       "   any Care Area and press OK.\n\n"\+
       "2. From the Drug List select BASIC and then press OK.\n\n"\+
       "3. Select Primary Bag and press OK.\n\n"\+
       "4. For Mode select mL/hr and press OK.\n\n"\+
       "5. Set the Rate to 100mL/hr and press OK.\n\n"\+
       "6. Set VTBI mL to 13.3 mL and press OK.\n\n"\+
       "When ready to continue, press 'Next Step'..." | small


prompt "Delay 20 seconds. Please wait..." | normal | 20
```

4.4 show Statement

Syntax: **show <string> | <style> | <filepath>**

UI Reference: Section 3.3.3

Description: The **show** statement displays a text string in a specified style, with a "**Show Picture**" or "**Play Video**" button beneath the text. When the button is pressed, the image or video file specified by **<filepath>** is passed to a multimedia application which then shows or plays the file.

All arguments are required. Text styles are the same as those specified for the **prompt** statement (see Section 4.3).

The filepath includes the filename of the image or video and all directories required to specify the path, up to but ***not*** including the name of the storage media (i.e. , the filepath will exclude '**mnt/sdcard**'). Similar to a string, the filepath may be enclosed in double-quotes but this would be for code readability only and is optional.

Supported file types are **.png** and **.jpg** for images, and **.mp4** and **.3gp** for video. The file type determines whether the button displayed by vPad-Check is "**Show Picture**" or "**Play Video**".

Examples:

```
show "Verify the equipment being serviced\n"\+
     "matches this photo of the defibrillator" | normal | \+
     "Datrend/vPad-ES/Instructions/hp_defib_1.jpg"


show "Play this video to see how to remove\n"\+
     "the rear panel of the equipment" | normal | \+
     "DCIM/Camera/VID_20131022_143820.mp4"
```

4.5 check Statement

Syntax: **check <string> | [mode]**

UI Reference: Section 3.3.4

Description: The **check** statement displays a text string along with radiobuttons and an optional text box for user input of data or comments. The **mode** argument configures the user interface as follows:

passfail Displays "**PASS**", "**FAIL**", "**N/A**" and "**SKIP THIS**" radiobuttons, with no text box

input Displays "**PASS**", "**FAIL**", "**N/A**", "**INFO ONLY**" and "**SKIP THIS**" radiobuttons, and a text box for input of comments or data

The string is the only required argument. The mode is optional and if not specified a default mode of **passfail** is used.

This statement saves information to the Test Record. The radiobutton which is pressed is recorded as the test result for the Checklist step. Contents of the comments/data text box, if any, are saved as the test data for the step. The **<string>** of the statement is also saved, to give the recorded data and result some context.

Examples:

```
check  "Chassis and housing...\n\nCheck general condition"

check  "Check power cord, plug and strain relief"

check  "Indicators and displays...\n\nCheck operation" | input

check  "Controls and switches...\n\nCheck operation" | input

check  "Check patient cables/leads" | input

check  "Test alarms\n\nand audible signals" | input
```

4.6 limtst Statement

Syntax:

```
limtst <string> | <UOM> | <comparison> | <X> | [Y] | [tol_type]
```

UI Reference: Section 3.3.5

Description: This statement encodes a "limit test", in which a user is prompted to input numerical data such as a measurement or a test result, and the entered value is then compared to one or two boundary values to obtain a result of "PASS" (measurement within limits) or "FAIL" (measurement outside of limits).

Arguments of the limit test statement are:

<string> This is a required argument. This message is intended to provide instruction to the user as to the quantity to be measured or tested and then entered into the "Test Data" field of the UI.

<UOM> This is the units of measure and it a required argument. This string is displayed to the right of the "Test Data" field and provides further information as to the quantity being measured or tested.

<comparison> This is a required argument which specifies the type of comparison performed by the limit test. The defined values for this parameter are:

min Compare entered data to a minimum threshold. If the value entered by the user is greater than or equal to threshold **X**, the test result is assigned a "PASS". If the entered data is less than **X**, the test result is "FAIL".

max Compare entered data to a maximum threshold. If the value entered by the user is less than or equal to threshold **X**, the test result is assigned a "PASS". If the entered data exceeds **X**, the test result is "FAIL".

range Compare entered data to an acceptance range. If the value entered by the user falls within the range set by lower limit **X** and upper limit **Y**, the test result is assigned a "PASS". If the value falls outside the range of **X** to **Y**, the test result is

"FAIL".

target Compare entered data to an acceptance range which is defined by an expected value or "target" **X** and a +/- tolerance **Y**. If the value entered by the user falls within the range defined by target **X** and tolerance **Y**, the test result is assigned a "PASS". If the value falls outside the range defined by **X** and **Y**, the test result is "FAIL".

<X> This is a decimal number and it is a required argument for all limit tests. According to the **comparison** selected, the **X** value will set the threshold for a **max** or **min** test, the lower bound for a **range** test, or the expected value of the entered data for a **target** test.

[Y] This is a decimal number and it is required only for **range** or **target** tests. According to the **comparison** selected, the **Y** value will set the upper bound for a **range** test or the +/- tolerance for a **target** test.

[tol_type] This argument selects the tolerance "type" and is required only for **target** tests. The argument has two defined settings, as follows:

units The tolerance **Y** is represented in the same units of measure as the target value **X**. In this case, the acceptance range for entered data will span **(X - Y)** to **(X + Y)**.

percent The tolerance **Y** is a percentage of the target value **X**. In this case, the acceptance range for entered data will span **(X - dX)** to **(X + dX)**, where **dX = (Y * X)/100**.

The **limtst** statement saves information to the Test Record. The value entered by the user is saved along with the "PASS" or "FAIL" result of the limit comparison. The **<string>** of the statement is also saved, to give the recorded data and result some context.

Examples:

```
limtst "Check battery voltage exceeds 10V" | volts | min | 10

limtst "Test occlusion pressure alarm\n"\+
       "of infusion device.\n\n"\+
       "Check pressure does not exceed 17.5 psi"\+
       | psi | max | 17.5
```

```
limtst "Measure power output.\nCheck power = 200 to 400 uW"\+
       | microwatts | range | 200.0 | 400.0

limtst "Check ECG ratemeter reads 60 +/- 2 BPM"\+
       | BPM | target | 60 | 2 | units

limtst "Discharge defib at 100J.\nCheck energy = 100J +/- 15%"\+
       | Joules | target | 100 | 15 | percent
```

4.7 color Statement

Syntax: **color <rgb_setting>**

Description: Sets the foreground colour of text as displayed by **prompt**, **check**, **limtst** or **show** statements. **rgb_setting** is a 24-bit number in hexidecimal notation which specifies the relative mix of red, green and blue in the overall colour. The argument begins with a '#' character, followed by red, green and blue values with each component respectively ranging from 00 (black) to FF (maximum intensity).

By default, Checklists display text in white (**rgb_setting** = #FFFFFF). When required, the default colour can be changed by preceding a **prompt** or other statement with a **color** statement. The colour setting remains in effect until changed by a subsequent **color** statement.

Examples:

```
color  #FF0000
prompt "This text will appear in bright red"

color  #00FF00
prompt "This text will appear in bright green"

color  #0000FF
prompt "This text will appear in bright blue"

color  #FFFF00
prompt "This text will appear in bright yellow"

color  #F8F8A0
prompt "This text will appear in a softer yellow"

color  #FFC700
prompt "This text will appear in orange"

color  #0080FF
prompt "This text will appear in a medium blue"

// Reset text colour to default (white)
color  #FFFFFF
```

4.8 power Statement

Syntax: **power <state>**

UI Reference: This statement produces no display indication on the Android tablet. The state of the test receptacle can be determined by observing the indicator lamp next to the receptacle.

Description: Controls the state of the test receptacle of the vPad Analyzer Base Unit. The **state** of the receptacle may be set to either **on** or **off** respectively.

Examples:

```
// Apply power to equipment under test
power  on

// Power down the equipment under test
power  off
```

4.9 ecgsim Statement

Syntax: **ecgsim <wave_number> | [mode] | [delta_bpm]**

UI Reference: Section 3.3.6

Description: The **ecgsim** statement controls the vPad ECG simulator and provides a choice of user interface for data entry and test result selection.

Arguments of the **ecgsim** statement are:

<wave_number> This argument is required. It is an integer value (1, 2, 3... 30) which represents the waveform to be generated by the ECG simulator according to the table below.

No.	Waveform	Description
1	SQR 2Hz	Square wave, 2 Hz, 1 mV p-p
2	TRI 2Hz	Triangle wave, 2 Hz, 1mV p-p
3	PUL 4sec	'DC' pulse, +1 mV for 4 sec, 0 mV for 4 sec
4	PUL 30ppm	Pulse, +1mV for 63 msec, rate = 30 ppm
5	PUL 60ppm	Pulse, +1mV for 63 msec, rate = 60 ppm
6	SQR 1 KHz	Square wave, 1 KHz, 1 mV p-p
7	SIN 0.5 Hz	Sine wave, 0.5 Hz, 1 mV p-p
8	SIN 10 Hz	Sine wave, 10 Hz, 1 mV p-p
9	SIN 40 Hz	Sine wave, 40 Hz, 1 mV p-p
10	SIN 50 Hz	Sine wave, 50 Hz, 1 mV p-p
11	SIN 60 Hz	Sine wave, 60 Hz, 1 mV p-p
12	SIN 100 Hz	Sine wave, 100 Hz, 1 mV p-p
13	CMRR SQR 2Hz	Square wave, 2 Hz, 10mV common mode
14	CMRR PUL 4sec	DC pulse, 10mV common mode
15	CMRR SIN 0.5Hz	Sine wave, 0.5 Hz, 10mV common mode

No.	Waveform	Description
16	CMRR SIN 50Hz	Sine wave, 50 Hz, 10mV common mode
17	CMRR SIN60Hz	Sine wave, 60 Hz, 10mV common mode
18	CMRR SQR 1KHz	Square wave, 1KHz, 10mV common mode
19	NSR 30 BPM	ECG normal sinus rhythm, 30 BPM
20	NSR 60 BPM	ECG normal sinus rhythm, 60 BPM
21	NSR 120 BPM	ECG normal sinus rhythm, 120 BPM
22	NSR 180 BPM	ECG normal sinus rhythm, 180 BPM
23	NSR 240 BPM	ECG normal sinus rhythm, 240 BPM
24	NSR 300 BPM	ECG normal sinus rhythm, 300 BPM
25	VFIB	Arrhythmia, ventricular fibrillation
26	AFIB	Arrhythmia, atrial fibrillation
27	SVT	Arrhythmia, supraventricular tachycardia
28	VTACH	Arrhythmia, ventricular tachycardia
29	PVC	Arrhythmia, preventricular contraction
30	ASYS	Arrhythmia, asystole, random baseline

[mode] Sets the user interface mode of the ECG test as follows:

passfail Enables pass/fail radiobuttons on the user interface as shown in ***Figure 24*** on page 24. This is the default if a **mode** is not specified in the statement.

input Enables pass/fail radiobuttons on the user interface with a text box for input of comments or data (see ***Figure 25***).

bpm Useable with NSR simulations #19 to #24 only. Enables a limit test which implicitly uses the BPM rate of the specified simulation as the target value (see ***Figure 26***). If this **mode** is selected for a non-NSR simulation, then the **passfail** mode will be applied by default.

[delta_bpm] Sets the pass/fail tolerance for **bpm** mode. If not specified, a default tolerance of ±2 BPM is applied.

The **ecgsim** statement saves information to the Test Record. The ECG waveform, the pass/fail result and comments or data entered by the user are recorded.

Examples:

```
// Set simulator to SQR 2Hz, simple pass/fail test
ecgsim   1

// Set simulator to AFIB, enable input of comments/data
ecgsim   26 | input

// Set simulator to ECG 60BPM, enable BPM rate test with
// default tolerance of +/- 2 BPM
ecgsim   20 | bpm

// Set simulator to ECG 180BPM, enable BPM rate test with
// tolerance of +/- 3 BPM
ecgsim   22 | bpm | 3
```

4.10 run Statement

Syntax: **run <app_name> | [autosequence]**

UI Reference: Section 3.3.7

Description: The **run** statement launches a vPad App from a Checklist. The App returns its results as a formatted text file, which is then copied into the vPad-Check Test Record.

Arguments of the **run** statement are:

<app_name> This is the vPad application to be launched by vPad-Check. This argument is required and may be **vPad-AS**, **vPad-A3** or **vPad-Cal**. Future vPad Apps may be callable from vPad-Check via the **run** statement, and the same syntax will generally apply to such future Apps.

[autosequence] This is the AutoSequence number (1, 2, 3...) to be executed by **vPad-AS** or **vPad-A3**. It is optional but should be specified. If not specified, then AutoSequence #1 will be executed by default. This argument is not used when launching **vPad-Cal**. Future Apps may potentially use this argument to represent something other than an AutoSequence number, as will be specified when such Apps become available.

Examples:

```
// Do a safety test with vPad-AS; run autosequence #4
run   vPad-AS | 4

// Do a safety test with vPad-A3; run autosequence #16
run   vPad-A3 | 16

// Get test equipment information using vPad-Cal
run   vPad-Cal
```

4.11 Remote Control Statements

4.11.1 xopen Statement

Syntax: **`xopen <target_device> | <address> | [setup_delay]`**

UI Reference: Section 3.3.8.2

Description: The **`xopen`** statement opens a connection to a remote test instrument by means of a corresponding XBUS Smart Cable (XSC) interface.

This statement must be the first remote control statement of a Checklist and as such the **`xopen`** statement will proceed any **`xctrl`** or **`xtest`** statements that are used. If a Checklist includes a second **`xopen`** statement, it will effectively close the first connection and open a new one; in this way a Checklist may interface with multiple testing devices within the same procedure, although with only one device at a time.

Each XSC adaptor is designed to interface vPad to a specific **`target_device`**. An exception is the MBOX interface (part no. 7400-105 (wireless) or -106 (serial)) which may be used for basic send/receive communication with any test instrument that supports remote control operation through a serial port.

The arguments of the **`xopen`** statement are:

`target_device` This is the name or ID of a test device to be controlled by the Checklist. The name is arbitrary, but it is required and must be specified in the statement. Typically, **`target_device`** will be the model of the test instrument or simulator which is under control, or a nickname or other ID which corresponds to the remote device. Since this argument is really a string, it can be optionally enclosed in double-quotes for better readability with a source code editor.

`address` This is a single alphabetic character (A, B, C... Z, a, b, c... z) which specifies the XBUS address of the XSC interface. The **`address`** character may be found in user documentation which is provided with the XSC interface. This is a required argument.

`[setup_delay]` This argument is optional and corresponds to the maximum amount of time needed for the XSC to set up the communication interface on the target device side. This delay is needed primarily for wireless

interfaces which may require slightly longer to initialize, depending on distance and radio environment. The **xopen** statement waits until the **setup_delay** expires before confirming that the XSC interface and remote test device are on-line and ready to continue with the test. The **setup_delay** is in seconds and may be set to any decimal number greater than zero. For most applications, a delay of 1 to 3 seconds will be appropriate; refer to user documentation provided with the XSC adaptor if necessary. If not specified, a default delay of 2 seconds is used.

Examples:

```
// Example #1
// Open a connection to AMPS-1 simulator via XSC interface 7400-104
// Use default setup delay of 2 seconds
xopen   AMPS-1 | A

// Example #2
// Open a connection to MBOX universal adaptor 7400-106
xopen   MBOX | M | 1

// Example #3
// Open a connection to Phase3 defib tester via Phase3 XSC interface
xopen   Phase3 | P

// Example #4
// Open a connection to Impulse 4000 defib tester
// Wireless adaptor is in use, so allow a little longer for setup delay
xopen   Impulse 4000 | i | 3.5

// Example #5
// Open a connection to Phase3, using a MBOX for comm interface
xopen   Phase3 | M | 1
```

4.11.2 xctrl Statement

Syntax: **`xctrl <string> | <command> | [cmd_delay]`**

UI Reference: Section 3.3.8.2

Description: The **`xctrl`** statement causes a remote test instrument to execute one or more operations under the control of a corresponding XBUS Smart Cable (XSC) interface. If the XSC interface is MBOX, the **`xctrl`** statement does not control the remote device but is instead used to configure the MBOX itself. After the command is sent, the Checklist proceeds to the next step automatically. The communication connection to the XSC must be initialized by an **`xopen`** statement before **`xctrl`** is used.

Arguments of the **`xctrl`** statement are:

`<string>` This is a required argument. This message is displayed on the screen while data communications are in progress. Usually, this string will be a description of the effect the **`<command>`** argument will have on the remote device which is being controlled.

`<command>` This is a required argument and its use will depend on whether MBOX or a device-specific XSC is being used to interface with the remote test instrument.

If MBOX is in use, the **`<command>`** configures the communication connection from MBOX to the test device (for example, setting the baud rate, character transmission delay and other serial port settings). In this case, the **`<command>`** is ***not*** sent to the test device but is used by MBOX itself. Refer to MBOX user manual for further details.

For other device-specific XSC, the **`<command>`** causes one or more control messages to be sent from the XSC to the test device to configure the device or cause the device to perform a specific function. It is possible the **`<command>`** may be passed directly to the test device without modification, but this is usually not the case. Refer to the user manual for the particular XSC for more information.

[cmd_delay] This is an optional argument which causes the Checklist to delay before automatically proceeding with the next step, thereby allowing a remote test device some time to execute the required operation(s). The delay is also useful in enabling the user to confirm correct operation of the test device, by means of the descriptive **<string>** which is shown on the screen during data communications. The **cmd_delay** is in seconds and may be set to any integer number greater than zero (1, 2, 3...). For many applications, a delay of 2 seconds will be suitable; refer to user documentation provided with the XSC adaptor if necessary. If not specified, a default delay of 2 seconds is used.

Examples:

```
// Example #1
// Make AMPS-1 simulator produce a static IBP level of -10 mmHg
// XSC adaptor is use is part no. 7400-104 for AMPS interface
// The control command is "161"
// Use default command delay of 2 seconds
xctrl   "Setting IBP static pressure\nto -10 mmHg..." | 161


// Example #2
// XSC adaptor is use is MBOX part no. 7400-106
// Set the baud rate of the MBOX serial port to 2400 baud
xctrl   "Set baud rate to 2400..." | }br1{ | 1
// Set MBOX character transmission delay to 100 msec
xctrl   "Set TX delay to 100msec..." | }td4{ | 1
// MBOX is now set up to communicate with remote test device
```

4.11.3 xtest Statement

Syntax: **`xtest <string> | <command> | [pass_delay]`**

UI Reference: Section 3.3.8.2

Description: The **`xtest`** statement causes a remote test instrument to execute a test procedure or a measurement function under the control of a corresponding XBUS Smart Cable (XSC) interface. If the XSC interface is MBOX, the **`xtest`** statement is used to send control messages to the test device and to receive measurement data sent by the device. The communication connection to the XSC must be initialized by an **`xopen`** statement before **`xtest`** is used.

Arguments of the **`xtest`** statement are:

`<string>` This is a required argument. This message is displayed on the screen while the remote test device is performing the requested test or measurement function. Usually, this string will be a description of the test or measurement that is initiated through the **`<command>`** argument.

`<command>` This is a required argument and its use will depend on whether MBOX or a device-specific XSC is being used to interface with the remote test instrument.

If MBOX is in use, the **`<command>`** is parsed by MBOX into a *control message* and a *protocol parameter*. The control message is sent by MBOX to the test device. Depending on the protocol extracted from the **`<command>`**, MBOX may wait until the test device returns a response, or alternatively MBOX may signal the Checklist to advance to the next step automatically. In the case where test data is returned by the remote device, the data is displayed and is then passed or failed manually by the user (see on page). Refer to MBOX user documentation for further information and application notes.

For other device-specific XSC, the **`<command>`** causes one or more control messages to be sent from the XSC to the test device to make the device to perform a specific test or measurement. The XSC then monitors the status of the test device as the test or measurement is in progress. Once test data are obtained from the remote device, the XSC

will return a result to the Checklist which may include pass/fail information. In the event the test result is assigned a "PASS" by the XSC, the Checklist may optionally advance to the next step automatically. Refer to the user manual for the particular XSC for more information.

[pass_delay] This argument is optional and is usable only with XSC interfaces that provide automatic pass/fail of test results obtained from a remote device. This argument has no effect in MBOX interface applications.

For XSC which pass or fail test results automatically, a time delay may be specified by this argument which takes effect when the test result is a "PASS". In this case, the Checklist will advance to the next step automatically after the **pass_delay** expires. The intent of the delay is to provide sufficient time for the user to recognize that the result obtained from the test device has been passed. The **pass_delay** is in seconds and may be set to any integer number greater than zero (1, 2, 3...). Typically, a delay of about 5 seconds should be adequate. If the test result is a "FAIL", or if no **pass_delay** is specified in the **xtest** statement, then the user must press the "**Next Step**" button to make the Checklist advance.

The **xtest** statement will save information to the Test Record if a test result is obtained from the remote test device. The data obtained from the device and the pass/fail result are recorded. The **<string>** of the **xtest** statement is also saved, to give the recorded information some context.

Examples:

```
// Example #1:
// Check output energy of a defibrillator at 100 Joules, using a test instrument
// Smart Cable in use is XSC interface for Datrend Systems Phase3 analyzer
// The Smart Cable will automatically pass or fail the energy result from Phase3
prompt   "Set defib energy to 100J\nand charge."
xtest    "Discharge defib.\nVerify energy is 100J+/-15%" | DE,1,100,15 | 5

// Example #2:
// Check output energy of a defibrillator at 100 Joules, using a test instrument
// Smart Cable in use is a MBOX, connected to a Datrend Systems Phase3 analyzer
// The user must pass or fail the energy measurement data returned by Phase3
prompt   "Set defib energy to 100J\nand charge."
xtest    "Discharge defib.\nVerify energy is 85J to 115J" | DEHT -999
```

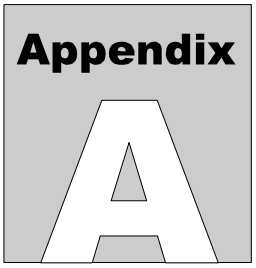
Chapter 5

5 Upgrades and Maintenance

Software Applications (Apps) may be upgraded from time-to-time to provide enhanced features or improvements. Contact Datrend or your local dealer for information on these upgrades.

In some cases, the changes made to an App may be to correct operational issues that have come to the attention of DSI. These shall generally be referred to as maintenance upgrades. In other cases, the changes may be to improve performance or add features which would enhance the overall functionality of the App. The decision to apply a charge to any upgrade shall be solely at the discretion of DSI.

Datrend Systems Inc.
Unit 130 - 4020 Viking Way
Richmond, BC • CANADA • V6V 2L4
Tel 604.291.7747 or 800.667.6557 • Fax 604.294.2355
e-mail **customerservice@datrend.com**
www.datrend.com



APPENDIX A. Example Checklists

Below is a listing of the "Demo Checklist" which is included with the vPad-Check App. The Demo Checklist illustrates how some of the more important keywords of the vPad Programming Language (including **prompt**, **check** and others) are used. Many other Checklist examples are provided with the vPad-Check App. Checklists may be viewed or modified with any word processor or text editor, including the text editor installed on the Android tablet.

```
// FILE: Demo Checklist.vpl
// Date: October 9 2013
// Version: 1.0
//
// A demo checklist providing examples of VPL statements, including
// "prompt", "check", "limtst" and others.

prompt  "DEMONSTRATION\nCHECKLIST" | bold

prompt  "A procedure for equipment inspection and/or maintenance can be "\+
        "implemented as a vPad 'Checklist'.\n\n"\+
        "A Checklist is made up of steps which are called 'statements'.\n\n"\+
        "This demo shows some examples of the statements which are available "\+
        "for creating Checklists, and how these work..." | medium

prompt  "'Prompt' statements are typically used\n"\+
        "to display messages or instructions"

prompt  "Prompts can be displayed\n"\+
        "in BOLD font, like this..." | bold

prompt  "...or in RED font\n\n(useful for warnings)" | red

prompt  "Prompts can be accompanied\n"\+
        "by a 'bell' sound..." | bell

prompt  "...and important messages can be\n\n"\+
        "emphasized with an 'alert' sound" | alert

prompt  "The 'medium' font, shown here, can be used in those situations "\+
        "where there is a need to display somewhat more information "\+
        "than will easily fit on the screen using the 'normal' font." | medium

prompt  "This is the 'mmono' font. It is slightly smaller than 'medium' "\+
        "font and it is uniformly-spaced. This is useful if you want to "\+
        "display some information in a tabular layout, for example, like "\+
        "this:\n\n\n"\+
        "               Column 1          Column 2\n"\+
        "                Item 1            Item A\n"\+
        "                Item 2            Item B\n"\+
        "                Item 3            Item C\n" | mmono
```

```
prompt  "A 'prompt' statement can also use 'small' font, like this. This is "\+
        "useful for displaying detailed instructions, for example:\n\n"\+
        "1. First, do this step.\n\n"\+
        "2. Next, do this step.\n\n"\+
        "3. Next, do this step.\n\n"\+
        "4. Finally, do this step. Since the small font is uniformly\n"\+
        "   spaced, it is possible to created indented paragraphs\n"\+
        "   such as the one shown here.\n\n"\+
        "When ready to continue, press 'Next Step'..." | small

color   #80FF80
prompt  "Prompts default to displaying\ntext in white.\n\n"\+
        "This default can be changed with a\n'color' statement, as shown here. "

color   #FFFFFF
prompt  "A timer can be attached to a prompt, as shown above. When the timer counts "\+
        "down to zero, the checklist will automatically proceed to the next step. "\+
        "If you do not want to wait for the timer to reach zero, you can press the "\+
        "'Next Step' button anytime.\n\n"\+
        "Prompts with a timer can be used to flash a quick message, or make a checklist "\+
        "delay until some other process, such as a test, has finished. Either way, a "\+
        "timer is useful for adding more automation to your checklists."\+
        | medium | 30

check   "The 'check' statement is used for checking\n"\+
        "or inspecting items. The statement has two\n"\+
        "forms. This is the basic pass/fail version." | passfail

prompt  "Next are some examples of typical\n"\+
        "pass/fail 'check' statements..."

check   "Check equipment for physical damage" | passfail

check   "Check line cord of equipment" | passfail

check   "Check power plug of equipment" | passfail

check   "The 'check' statement can also be set to\n"\+
        "accept user input, as shown here" | input

prompt  "Next are some examples of 'check'\n"\+
        "statements that accept user input..."

check   "Check indicators & displays" | input

check   "Check fuses & enter ratings" | input

check   "Enter the software version\nof the patient monitor" | input

prompt  "The 'limtst' statement is used for tests which will involve measurements.\n\n"\+
        "You enter the measurement into a 'Test Data' box, then you press the "\+
        "'Check Limits' button. The data you entered is compared to upper and/or lower "\+
        "limits which are defined in the 'limtst' statement. The measurement is then "\+
        "assigned a pass or fail, based on the limit(s) which have been set.\n\n"\+
        "Press 'Next Step' to see a 'limtst' example for a defib..." | medium

limtst  "Set defib energy to 50J and discharge.\n"\+
        "Measure output energy with defib analyzer\n"\+
        "and enter below as 'Test Data'.\n"\+
        "Measured energy should be 50J+/-15%" | Joules | target | 50 | 15 | percent

prompt  "The message on the previous screen has a lot of text. But, less "\+
        "information can be displayed if the user knows what they should do "\+
        "when asked to perform a test and then enter the measurement result.\n\n"\+
        "For example, press 'Next Step' to see a simpler (and more typical) "\+
```

```
        "example of how to apply the 'limtst' statement..." | medium

limtst  "Measure power output.\n\nCheck power = 200 to 400 uW"\+
        | microwatts | range | 200 | 400

show    "The 'show' statement can be used to\n"\+
        "display a photograph..." | normal |\+
        "Datrend/vPad-ES/Instructions/hp_defib_1.jpg"

show    "The 'show' statement can also be\n"\+
        "used to play a video..." | normal |\+
        "Datrend/vPad-ES/Instructions/diagnostic_ecg_video.mp4"

show    "A smaller font can be used for the message displayed above "\+
        "the 'Show' button, in the event there is a need to provide "\+
        "a detailed explanation for the image or video that will be "\+
        "displayed..." | medium |\+
        "Datrend/vPad-ES/Instructions/hp_defib_3_ecg.jpg"

prompt  "The next step will demonstrate the\n"\+
        "'power' statement, which will set\n"\+
        "the test receptacle ON"
power   on

prompt  "The next few steps will demonstrate operation of the ECG "\+
        "simulator, via the 'ecgsim' statement.\n\n"\+
        "The first example test will be a SQR 2Hz performance wave, with "\+
        "optional test result input via pass/fail buttons..." | medium
ecgsim  1

prompt  "The next example is a AFIB arrhythmia, with optional input of "\+
        "comments enabled, along with pass/fail buttons" | medium
ecgsim  26 | input

prompt  "For NSR waveforms, a tolerance in BPM can be optionally specified "\+
        "for the rate, which is predefined for the waveform chosen. Press "\+
        "'Next Step' to see this example..." | medium
ecgsim  20 | bpm | 3

prompt  "Next, this demo will perform a\n"\+
        "quick safety test by running\n"\+
        "the vPad-AS application..."
run     vPad-AS | 5

prompt  "Other statements can remotely control external test instruments and "\+
        "download test results directly to vPad.\n\n"\+
        "To make use of this capability, you will need an accessory 'Smart Cable' "\+
        "to connect the external tester or simulator of interest to vPad.\n\n"\+
        "To see an example of how this works, try the 'Remote Control Demo - AMPS-1' "\+
        "checklist." | medium

prompt  "That's the end of the show.\n\nSee you next time!"

// Make sure the test receptacle is OFF
power   off

// End of script
```

Below is a listing of "Remote Control Demo - AMPS-1". This example illustrates how some of the remote control statements of vPL are used.

```
// File: Phase 3 Demo with MBOX.vpl
// Date: Oct. 25 2013
// Rev.: 1.0
//
// Demo checklist using a Phase 3 with MBOX interface for defib
// energy and other measurements. Energies tested are 5, 10, 50, 100,
// 200 and maximum energy (charge time test). Pass/fail is by manual
// entry, based on displayed acceptance ranges (15% tolerances).
// Cardioversion/sync is also tested.

prompt  "DEMO OF PHASE 3\nREMOTE CONTROL VIA MBOX" | bold

prompt  "Connect Phase 3 serial port to vPad\n"\+
        "via MBOX adaptor (P/N 7400-106)."

prompt  "Apply power to Phase 3"

prompt  "Set Phase 3 to REMOTE mode"

prompt  "During this procedure, if you need to cancel a test that has "\+
        "been started on the Phase 3, remember to press the CANCEL button "\+
        "at the lower right corner of the Phase 3's LCD display" | medium

prompt  "Connect paddle plates to Phase 3\n"\+
        "or use hands-free cable"

// Reset MBOX interface to defaults (9600 baud; no delay; EOC = CR)
xopen   "Phase 3" | M | 1

// Set Phase 3 ECG simulator to Coarse VFIB
// Command sent to Phase 3 is "ECGDJ10"
// Protocol = -2 causes this checklist to go to the next step
// automatically when Phase 3 returns "*" handshake response
xtest   "Setting ECG to Coarse VFIB..." | ECGDJ10 -2

// Apply power to defib under test
power   on

// Prompt user to set the defib energy for the test
prompt  "Set defib to 2 Joules"

// Get Phase 3 to perform defib energy test, low range, return data = text
// "DELT" command returns a string like this: "ENERGY=ddd.dJ"
xtest   "Discharge defib.\nVerify energy = 1.7 to 2.3J" | DELT

// Repeat for 10J energy
prompt  "Set defib to 10 Joules"
xtest   "Discharge defib.\nVerify energy = 8.5 to 11.5J" | DELT

// Repeat for 50J energy (high range test)
prompt  "Set defib to 50 Joules"
xtest   "Discharge defib.\nVerify energy = 43 to 58J" | DEHT

// Repeat for 100J energy (high range test)
prompt  "Set defib to 100 Joules"
xtest   "Discharge defib.\nVerify energy = 85 to 115J" | DEHT

// Repeat for 200J energy (high range test)
prompt  "Set defib to 200 Joules"
xtest   "Discharge defib.\nVerify energy = 170 to 230J" | DEHT
```

```
// Charge time test at max energy (high range test)
prompt  "Set defib to run on battery power, and select maximum energy"
xtest   "Charge up & discharge.\nCheck charge time < 10 sec" | DCHT

// Set Phase 3 ECG simulator to Coarse AFIB
xtest   "Setting ECG to Coarse AFIB..." | ECGDA10 -2

// Cardioversion test (high range test; sync to Q-wave)
prompt  "Set energy to 100J and\nenable SYNC"
xtest   "Discharge and check\nSYNC is 20 to 64 msec" | DSHQT


prompt  "End of Phase 3 demo"

// Power down the equipment under test
power   off

// End of checklist
```

vPad-Check OPERATORS MANUAL